2008-03

Proof of concept integration of a single-level service-oriented architecture into a multi-domain secure environment

Gilkey, Craig M.

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

PROOF OF CONCEPT INTEGRATION OF A SINGLE-LEVEL SERVICE-ORIENTED ARCHITECTURE INTO A MULTI-DOMAIN SECURE ENVIRONMENT

by

Craig M. Gilkey

March 2008

Thesis Advisor:	Cynthia E. Irvine
Second Reader:	Randy W. Maule

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2008	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE Proof of Concept Integration of a Single-Level Service-Oriented Architecture into a Multi-Domain Secure Environment			**5. FUNDING NUMBERS**
6. AUTHOR(S) Craig M. Gilkey			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

Service-Oriented Architecture (SOA) software has revolutionized data interchange in the business world. A SOA software platform integrates independent, unrelated applications into a common architecture, thereby introducing data reuse, interoperability, and loose coupling between the services involved. The U.S. Navy is currently experimenting with a SOA-based research portal called TACFIRE, or Tactical Applications for Collaboration in FIRE (FORCEnet Innovation and Research Enterprise). TACFIRE provides a set of lightweight, XML-based web services derived from the Oracle Collaboration Suite (OCS) 10g SOA. Such web services operating across multiple security domains would provide additional advantages, including improved intelligence aggregation, and real-time collaboration between users in different security domains. However, current TACFIRE implementations provide no multi-domain functionality between different classification levels.

To date, the incorporation of a SOA software suite into a multilevel secure environment has neither been designed nor implemented. This project has explored how a SOA software suite could be integrated into a multilevel environment. The OCS 10g has been configured to run within the Monterey Security Architecture (MYSEA) multilevel testbed. This thesis addresses DoD requirements for building enterprise-level computing architecture capable of providing a full range of information services at all major security classifications and information handling caveats.

14. SUBJECT TERMS Monterey Security Architecture, Service-Oriented Architecture, SOA, FORCEnet, TACFIRE, Oracle Collaboration Suite 10g			**15. NUMBER OF PAGES** 158
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

PROOF OF CONCEPT INTEGRATION OF A SINGLE-LEVEL SERVICE-ORIENTED ARCHITECTURE INTO A MULTI-DOMAIN SECURE ENVIRONMENT

Craig M. Gilkey
Lieutenant Junior Grade, United States Navy
Computer Science - B.S., Old Dominion University, 2004

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS AND OPERATIONS

from the

NAVAL POSTGRADUATE SCHOOL
March 2008

Author: Craig M. Gilkey

Approved by: Cynthia E. Irvine
Thesis Advisor

Randy W. Maule
Second Reader

Dr. Dan C. Boger
Chairman, Department of Information Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Service-Oriented Architecture (SOA) software has revolutionized data interchange in the business world. A SOA software platform integrates independent, unrelated applications into a common architecture, thereby introducing data reuse, interoperability, and loose coupling between the services involved. The U.S. Navy is currently experimenting with a SOA-based research portal called TACFIRE, or Tactical Applications for Collaboration in FIRE (FORCEnet Innovation and Research Enterprise). TACFIRE provides a set of lightweight, XML-based web services derived from the Oracle Collaboration Suite (OCS) 10g SOA. Such web services operating across multiple security domains would provide additional advantages, including improved intelligence aggregation, and real-time collaboration between users in different security domains. However, current TACFIRE implementations provide no multi-domain functionality between different classification levels.

To date, the incorporation of a SOA software suite into a multilevel secure environment has neither been designed nor implemented. This project has explored how a SOA software suite could be integrated into a multilevel environment. The OCS 10g has been configured to run within the Monterey Security Architecture (MYSEA) multilevel testbed. This thesis addresses DoD requirements for building enterprise-level computing architecture capable of providing a full range of information services at all major security classifications and information handling caveats.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

This thesis is dedicated to the memory of my late father, Harry E. (“Bud”) Gilkey, Jr. (Mechanical Engineering-Master of Science, University of Tennessee, and Electrical Engineering-Master of Science, University of Kentucky). We miss you, Dad.

This thesis is also dedicated to my wife, Melinda, and our two beautiful children, Sean and Veronica, who supported me while I was working late trying to finish this project.

I would like to thank my advisor, Dr. Cynthia Irvine, for the time and effort she has provided throughout this project. Additionally, I would like to thank my second reader, Randy Maule, for the technical guidance he provided regarding both TACFIRE and the Oracle Collaboration Suite 10g software suite. I would also like to thank Charles Prince, Jean Khosalim, Phil Hopfner, and Thuy Nguyen for providing technical expertise concerning the MYSEA testbed.

This material is based upon work supported by the National Reconnaissance Office. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the N R O.

THIS PAGE INTENTIONALLY LEFT BLANK

ACRONYMS AND ABBREVIATIONS

AKO	Army Knowledge Online
BEA	Business Enterprise Architecture
C2PC	Command and Control Personal Computer
C4I	Command, Control, Communications, Computers, and Intelligence
CENTRIX	Combined Enterprise Regional Information Exchange System
CISR	Center for Information Systems Security Studies and Research
COTS	Commercial Off The Shelf
DAC	Discretionary Access Control
DAV	Distributed Authoring and Versioning
DISA	Defense Information Systems Agency
DKO	Defense Knowledge Online
DoD	Department of Defense
FIRE	FORCEnet Innovation & Research Enterprise
GCCS	Global Command and Control System
GIG	Global Information Grid
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol over Secure Socket Layer Encryption
IDG	International Data Group
IETF	Internet Engineering Task Force
IMAP	Internet Message Access Protocol
IP	Internet Protocol
IT	Information Technology

J2EE	Java Enterprise Environment
JSR 168	Java Specification Request 168
LDAP	Lightweight Directory Access Protocol
MAC	Mandatory Access Control
MLS	Multilevel Secure
MYSEA	Monterey Security Architecture
NATO	North Atlantic Treaty Alliance
NIPRnet	Unclassified but Sensitive Internet Protocol Router Network
NKO	Navy Knowledge Online
NMCI	Navy-Marine Corps Intranet
NNWC	Naval Network Warfare Command
NOC	Network Operating Center
OC4J	Oracle Container for J2EE
OCS	Oracle Collaboration Suite
OEF	Operation Enduring Freedom
OIF	Operation Iraqi Freedom
OS	Operating System
PBX	Private Branch Exchange
PDA	Personal Data Assistant
PEO	Program Executive Officer
POP	Post Office Protocol
RAM	Random Access Memory
RFC	Request For Comments
RTC	Real-Time Collaboration
SAK	Secure Attention Key
SATCOM	Satellite Communications
SIPRNet	Secret Internet Protocol Router Network
SMTP	Simple Mail Transport
SOA	Service-Oriented Architecture

SOAP	Service-Oriented Architecture Protocol
SSL	Secure Socket Layer Encryption
SSO	Single Sign-On
STOP	Secure Trusted Operating Program
TACFIRE	Tactical Applications for Collaboration in FIRE (FORCEnet Innovation & Research Enterprise)
TCBE	Trusted Computing Based Extension
TCP	Transmission Control Protocol
TPE	Trusted Path Extension
UDDI	Universal Description, Discovery, and Integration
VoIP	Voice Over IP
WebDAV	World Wide Web Distributed Authoring and Versioning
WSDL	Web Services Description Language
WSRP	Web Services for Remote Portals
XML	Extensible Markup Language
XMPP	Extensible Messaging and Presence Protocol
XTS	XML Transformation Server

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. MOTIVATION

The Department of Defense (DoD) has singled out Service-Oriented Architecture (SOA) implementation as the best way to 'fulfill the requirements of a net-centric environment [1].' The Defense Knowledge Online portal (DKO), the future 'single-entry point' for all service portals in the DoD, is based entirely on SOA [2]. In the business world, SOA provides organizations with interoperable, loosely coupled services that can be realigned and adapted to meet changes in the market [3]. For the DoD, there are additional advantages associated with implementing SOA. Transforming the DoD's inefficient, inflexible (and sometimes duplicate) legacy based applications into scalable, SOA-based software will reduce network overhead, decrease IT infrastructure spending, and unify future DoD SOA-based software under a single set of standards and protocols [1], [4], [5]. For these reasons, the DoD's intended future business enterprise architecture (BEA) is based entirely on SOA [2], [5].

A research portal (e.g., TACFIRE) based on SOA web services has been successfully tested and implemented in the fleet [6]. However, current implementations of this portal provide no cross-domain functionality between different classification levels across DoD networks [6]. A cross domain SOA would provide several advantages to the DoD: the aggregation of intelligence through file content management, and provision of SOA-based real-time collaboration applications (e.g., chat, web conferencing) to geographically disparate users at different classification levels.

The Monterey Security Architecture (MYSEA) project of the Naval Postgraduate School includes a working multilevel secure testbed [7]. The testbed integrates stateless commercial-off-the-shelf (COTS) hardware and software-based clients with specialized, high-assurance elements to enforce a unified, mandatory access control policy [7], [8], [9]. The motivation of this study is to integrate SOA-based software (and the benefits of such services) into the MYSEA environment.

B. PURPOSE

This thesis asked if a Service-Oriented Architecture (SOA) software suite could be integrated into the MYSEA environment. Through research regarding (a) Service-Oriented Architecture, (b) the Department of Defense's use of such software, (c) an existing SOA-based DoD research project (e.g., the TACFIRE portal), (d) the software platform it utilizes (Oracle Collaboration Suite 10g), and (e) high-assurance multilevel security architecture, it was determined that a SOA software suite could be integrated into a multilevel environment. This project set out to construct a proof of concept implementation to serve as a demonstration for the suggested concept. This involved the development of a functional test plan to specify (a) what SOA software suite to deploy, and (b) what SOA services to test. Preliminary tests of the proof of concept implementation were successful.

C. ORGANIZATION

This thesis is organized as follows:

- Chapter I provides an introduction to the motivation and purpose of this thesis.
- Chapter II provides background information on Service-Oriented Architectures (SOA), SOA software in the DoD, the TACFIRE research portal, the Oracle Collaboration Suite (OCS) 10g, and MYSEA.
- Chapter III describes specific goals of this thesis and the high level design used to accomplish these objectives.
- Chapter IV describes how the SOA software suite, the OCS 10g, was integrated into a simulated segment of the MYSEA multilevel testbed.
- Chapter V describes both the test design and the test results obtained in this experiment.
- Chapter VI provides an analysis of the experiment's test results, as well as a discussion regarding some of the design and configuration issues noted during installation and testing.
- Chapter VII summarizes the project, and makes suggestions for future work.

II. BACKGROUND

This chapter provides background information regarding specific subjects related to the incorporation of Service Oriented Architecture (SOA) software into the Monterey Security Architecture (MYSEA) environment. The first section discusses the origins, implementations, and benefits of SOA software in the business world. The second section discusses the relevance of incorporating SOA-based software into the Department of Defense. The third section discusses the development of the Navy's experimental SOA-based web portal, TACFIRE (Tactical Applications for Collaboration in FIRE [FORCEnet Innovation & Research Enterprise]). The fourth section describes the SOA software suite implemented by the TACFIRE portal, the Oracle Collaboration Suite 10g. The fifth section of this chapter gives a broad overview of the MYSEA project, and the sixth section is a summary of this chapter.

A. SERVICE-ORIENTED ARCHITECTURE (SOA)

Service-Oriented Architectures (SOA) have revolutionized application control, application reuse, and application interoperability in the business world. A SOA, by definition, is a *software architecture*: it is a set of statements that describes software components, and assigns functionality of the system to these components [10], [11]. This architecture is the 'system blueprint' for the organization's business model, and therefore resides at the highest (strategic) level of the organization. A SOA is specifically designed to increase the organization's flexibility in dealing with rapid changes to the business requirements of the organization itself. Implementing a SOA transforms the existing applications of an organization to provide a coherent, yet flexible and interoperable, architecture that can react and be modified when market shifts occur. In organizations that employ traditional enterprise-based Information Technology (IT) solutions, the ability to quickly adapt to market demands and/or conditions may be limited by the rigidity of the organization's applications themselves. If legacy-based applications cannot be modified quickly enough to meet the needs of a shifting global market, the organization's output may suffer. Figure 1 suggests how implementing a SOA might

provide a traditional enterprise-based organization 'agility' (or flexibility) when it is facing sudden market changes [8]. As market conditions shift, the organization's legacy applications cannot realign and adapt quickly enough to meet the demands of the market. Such 'realignment' typically involves either modification to the existing application code, or development of new applications to meet the needs of the market. Lacking the flexibility to transform rapidly, the organization's productivity diminishes. Figure 1 suggests that by implementing a SOA, the changes implemented will slowly transform the organization's applications to meet the current demands of the market. In a process known as 'orchestration,' a member of the organization (the SOA architect) can use the SOA to link, rearrange and even sequence the organization's services to meet new or existing market requirements [11], [12]. This process (orchestration) does not involve modification of the underlying application code [3], [11]. However, the SOA architect *does* need to be intimately familiar with the underlying business processes of the organization [11]. As shown in Figure 1, this gradual transformation (from legacy-based applications to a SOA) allows the organization to expand towards a state of 'flexible response.' This 'flexible response' is achieved through the orchestration of an organization's services to meet new business requirements as they arise [3], [10]. With the rapidly shifting market conditions of today's global economy, this flexibility provides the organization the ability to respond quickly to market transitions, thereby remaining competitive [4]. An independent online survey conducted by IDG Research Services Group indicated that the number of business organizations utilizing an enterprise-wide SOA has grown slightly year-by-year from 2005 to 2007 [13]. Each of the three surveys (2005, 2006, and 2007) polled over 1,000 IT professionals in various industries [13]. The number of businesses employing an enterprise-wide SOA was measured at 8% in 2005, at 16% in 2005, and at 21% in 2007 [13]. Although this is by no means a dramatic increase, it indicates that more businesses have transitioned from legacy based applications towards enterprise-wide SOA software since 2005 [13].

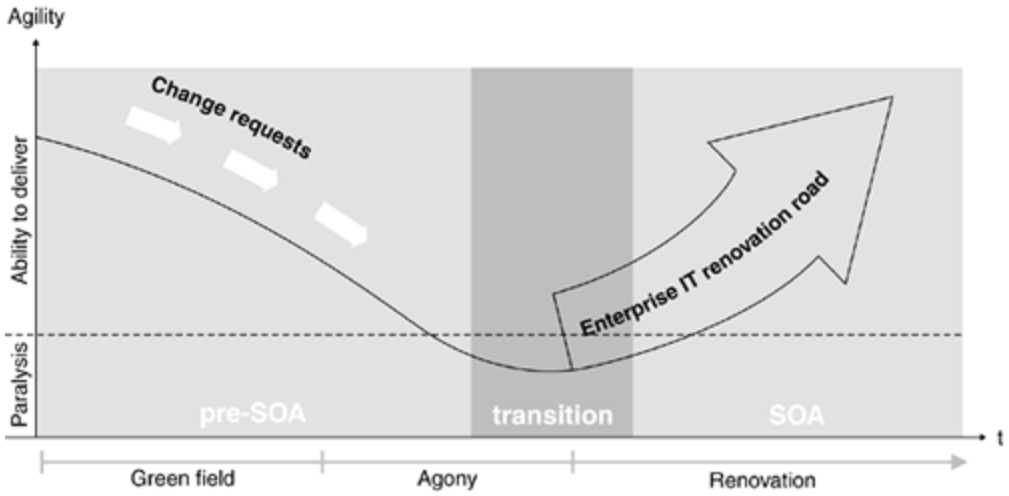

Figure 1. Service-Oriented Architecture is a key element of an enterprise IT renovation roadmap [After [11]].

Developments in programming languages, distribution technology, and business computing over the past 40 years have supported the emergence of Service-Oriented Architectures [11]. These evolutionary developments also contributed to the core fundamentals of the SOA: the application front end, the service, the service repository, and the service bus.

Figure 2 illustrates the four fundamental components of the SOA, and also displays the individual parts of the service component (the contract, the implementation [business logic and data], and the interface). The *application front end* is regarded as the owner of the business process. The application front end always starts a business process, and it always receives the results of the transaction conducted by the service [11]. A *service* is a software component with distinct, functional meaning, encapsulating a high-level (strategic) business concept [11]. Services deliver business functionality that the application front ends and other services need [3], [11]. In this context, 'business functionality' of the service is specifically designed for the client involved. This specification, designated as the *service contract*, stipulates the usage, constraints, and specific functionality of the service for a given client. The individual service must also be capable of interfacing with other services (via a *service interface*). The service contracts

of all of the services are held by the third component of the SOA, the *service repository*. The last major component, the *service bus*, provides a connection between the services and the application frontends [8].

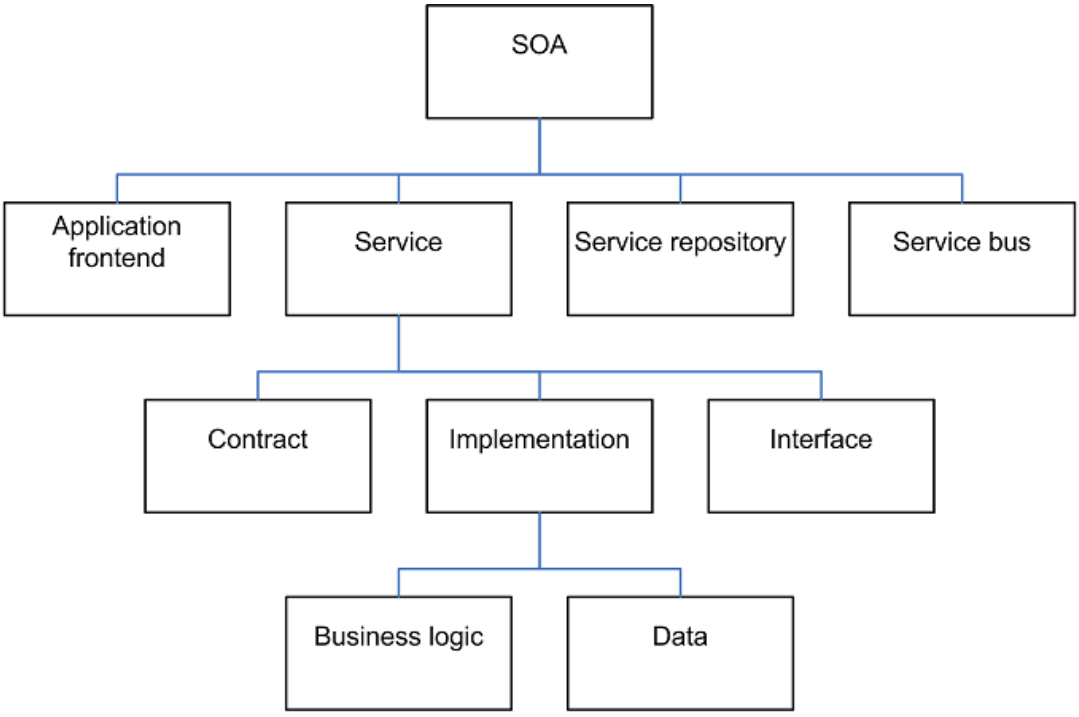

Figure 2. SOA key components [From [11]].

By design, the service is the most important component of the SOA. Although the application frontends are the most active components in a SOA, they are subject to frequent change, and, as a result, have a much shorter lifecycle than the services [11]. The service is also the one component providing reuse and interoperability for the SOA. By incorporating reuse at the macro (or service) level, businesses and organizations can respond more quickly to changing market conditions [3], [11]. Interconnections between existing, distributed IT assets and applications are simplified and improved through the use of well-defined, interoperable interfaces. The initial 'learning phase' to transition from existing legacy applications to SOA-based applications is quite expensive, particularly due to the immense complexity involved in modifying applications for a SOA. Once this learning phase is complete, however, the development costs are limited to application reuse or application extension. The transformed applications are both

mobile and interoperable. They can be distributed over a network, they can communicate with one another, and they can coordinate activity between two or more services. Multiple services in a SOA can be combined to create business applications [3], [11], [14].

Because of the interoperability of the SOA's services, newly-transformed legacy applications exchange data and participate in disparate business processes seamlessly. This 'software alignment' is extremely beneficial to an organization where multiple versions of application software, operating systems, and client hardware exist between various departments [14]. The applications incorporated into a SOA can be independent and unrelated, yet they share a common, congruent architecture [11]. This concept reinforces the existing advantages of the services (data reuse, interoperability, and loose coupling). Additionally, the tracking of version control issues is mostly limited to the client hardware components themselves, not the SOA software providing the services [1].

B. SERVICE-ORIENTED ARCHITECTURE (SOA) SOFTWARE IN THE DEPARTMENT OF DEFENSE

For the Department of Defense, a considerable amount of time will be required to transform existing legacy DoD systems into a SOA-based Business-Enterprise Architecture (BEA). 'BEA' is a business architecture term specific only to the DoD, and is defined as:

> The Business Enterprise Architecture (BEA) is the enterprise architecture for the Department of Defense's (DoD's) business information infrastructure and includes processes, data, data standards, business rules, operating requirements, and information exchanges. The BEA serves as the blueprint to ensure the right capabilities, resources and materiel are rapidly delivered to our warfighters through ensuring accurate, reliable, timely and compliant information across the DoD. The BEA achieves improved support to the warfighter through enabling streamlined processes, getting the Armed Forces what they need, where they need it, when they need it [2].

When asked how long such a transformation would take, the Chief Technical Officer of the DoD Business Mission Area replied, "it will take a generation [5]." Countless military-specific, legacy applications are spread out over various departments and

agencies. Individual software-based programs, such as Common Operational Picture (COP) software, vary in terms of interoperability between services. For example, in the Global Command and Control System (GCCS) program, there exists one subprogram for the Army (Global Command and Control System-Army) and one subprogram for the Navy (Global Command and Control System-Maritime) [15]. Both were developed with the same purpose: "to allow commanders to maintain topsight over the battlefield; collaborate with superiors, peers and subordinates over live data; and communicate their intent" [5], [16]. But currently, neither system is interoperable with the other in terms of sharing data [15].

Although this transformation will be both costly and lengthy, it may be worth the struggle. Since the end of World War II, the DoD hasn't changed its business model in terms of IT infrastructure [5]. Legacy IT components were built as 'stovepipes,' being neither interoperable nor interchangeable with other DoD applications and/or systems. As a result of this inefficiency, the DoD now spends almost 45% of its information technology budget on IT infrastructure costs alone [5]. Duplication or triplication of identical business processes is not uncommon; one legacy system might perform exactly the same process as another, yet support and bandwidth would be required for both [5]. Aside from a reduction in IT infrastructure spending, there are other reasons why such a large-scale transformation is necessary. Some key functions of the DoD, such as procurement and personnel management, are great candidates for conversion into 'services' to be shared by all branches and/or components [5]. Although an abundance of personnel management software exists in the DoD, there is little variation between the various service branches in the processes associated with personnel management itself. As the number of operations supporting the Global War on Terror increase, deployed operational forces will continue to demand increased bandwidth for data and applications. Transforming inefficient, inflexible (and sometimes duplicate) legacy applications into scalable, agile SOA-based software will reduce network overhead on an already bandwidth-limited DoD network infrastructure.

Both PEO C4I and the DoD I.T. Standards Registry provide specific guidance concerning SOA-based software in the DoD's Business Enterprise Architecture [4], [1].

The DoD I.T. Standards Registry requires the following protocols for DoD SOA: Universal Description, Discovery, and Integration (UDDI), Web Services Description Language (WSDL), Service-Oriented Architecture Protocol (SOAP), Lightweight Directory Access Protocol (LDAP), World Wide Web Distributed Authoring and Versioning (WebDav), Java Specification Request 168 (JSR 168), and Web Services for Remote Portals (WSRP) [4]. The Net-Centric Enterprise Solutions for Interoperability (NESI) overview document reiterates several SOA concepts:

> SOA promotes flexibility and reuse. This enables developers to compose complex software systems from clearly defined, implementation-neutral interfaces rather than through brittle implementation mechanisms such as tightly coupled, highly integrated applications… SOA isolates the specifics of data implementation from the service interface… Services are designed to be highly interoperable, loosely coupled, decentralized, and discoverable across the enterprise [1].

PEO C4I concludes that a Service-Oriented Architecture "best fulfills the requirements of a net-centric environment… (where) multiple clients and other services can access mission application functionality as a set of services [1]."

One focus of the DoD's BEA is to build a single, common portal known as the Defense Knowledge Online (DKO) Portal [5]. The SOA-based DKO Portal will eventually provide "a single user interface to government and industry for all (DoD) Enterprise IT services [2]." The DKO Portal will be a common, secure entry point for all areas of the DoD (service branches, agencies, and components), and securely link to existing portals, such as the Navy Knowledge Online (NKO) portal and the Army Knowledge Online (NKO) portal [5], [17].

There are numerous benefits associated with constructing a SOA-based portal onto the existing Department of Defense (DoD) network infrastructure. Simple, SOA-based web services like webmail, real time collaboration, and chat can be distributed uniformly between disparate, incongruent commercial off-the-shelf based client workstations [17]. Existing network infrastructure, client hardware and operating systems can be retained with very minor alterations. In an organization like the Department of the

Navy (supporting over 400,000 Navy-Marine Corps Intranet [NMCI] clients), this represents an enormous savings in terms of hardware reuse [18].

It should be noted that the cost to implement a web service based SOA with a limited number of applications still isn't cheap, as all of the legacy applications drafted for use in the SOA must be completely rewritten [19]. But a SOA providing web services is extremely advantageous for deployed, widely separated DoD units. Forward deployed commands could utilize the SOA services with a reduced number of servers, since the most application servers could be securely located at a Network Operating Center (NOC) in the continental United States. This limits security issues to the client's web browser and the application server located stateside [6]. The network overhead generated from SOA-based web services is somewhat lightweight, mostly because the individual web services are user-initiated. The amount of bandwidth needed by a SOA web service is directly correlated to current application demand: if the application is not requested by a user (or a group of users), bandwidth is not allocated to the application in question [11], [11]. Such 'smart-pull' characteristics are very beneficial for deployed units relying on limited bandwidth to communicate (e.g., deployed units relying solely on Satellite Communications (SATCOM) links to transfer data) [1], [5], [11], [15].

C. THE TACFIRE PORTAL

The U.S. Navy is currently experimenting with a SOA-based research portal called TACFIRE, or Tactical Applications for Collaboration in FIRE (FORCEnet Innovation and Research Enterprise). One of the initial goals of the TACFIRE research portal was to determine if a SOA-based architecture and associated web services were compatible with the existing legacy networks of the U.S. Navy [20], [21]. The TACFIRE project did not, however, intend to be a true, fleet-wide SOA (placing all of the Navy's applications under one SOA umbrella) [6], [21]. The scope of the TACFIRE project was limited to web-based applications that could be incorporated into the existing legacy architecture with few changes [6], [20], [21], [22].

TACFIRE utilizes a bundle of lightweight, XML-based web services, all while reusing the existing legacy network infrastructure, network hardware, client hardware and

client operating systems [6], [22], [23]. The TACFIRE portal employs a modified version of the Oracle Collaboration Suite 10g SOA software suite, a suite of enterprise-class real time collaboration, communication, and content management applications [6], [21], [22]. Deployed naval units can access a plethora of real-time web services in this service-oriented architecture, including such applications as Web Conferencing, Chat, Voice Over IP (VoIP), and WebDav-based File Content Services. Most importantly, the TACFIRE users can access the portal over a wide variety of U.S. and Coalition networks (including Unclassified, SIPRNet, and NATO networks). The TACFIRE portal employs Hyper-Text Transfer Protocol (HTTP) with Secure Socket Layer Encryption (SSL) to meet Defense Information Systems Agency (DISA) security requirements for web based applications [1], [5], [21], [24], [25].

The TACFIRE portal employs a 'dark-fiber, dumb network' philosophy. An ultra-thin client (a web browser) interacts with an enterprise-class server over a network [6]. In terms of security, this configuration significantly reduces the chance of compromise, since access to the network is limited to (a) the client's browser, and (b) the enterprise server itself [6]. The current server-client configuration in the fleet involves the exact opposite concept: a 'fat client' model. In a fat client network, the maintenance costs associated with servicing multiple, geographically separated servers are much higher than the maintenance required of a single, SOA-based enterprise server farm located stateside. However, current TACFIRE implementations provide no cross domain functionality between different classification levels across DoD networks [6], [20], [23]. Incorporating a SOA into a multilevel environment would provide this functionality, and such a 'cross domain' DoD SOA system would be advantageous for several reasons. The MLS ability to 'read down' could provide a SOA-based file content service with the ability to pool similar intelligence from different sensitivity levels. Intelligence aggregation could be improved tremendously in this manner, as mission critical data from lower sensitivity levels could be instantly aggregated with more sensitive data via a single, web-based multilevel aware SOA search application. Implementing a SOA server into each of the various DoD networks (NIPRNet, SIPRNet, CENTRIX, etc) would provide the benefits of the SOA-based services into each network individually. In this configuration, users

could utilize SOA-based real-time collaboration applications to chat or hold web conferences on any network classification to which they have access from a single, stateless COTS-based client [20], [22], [24], [25].

D. ORACLE COLLABORATION SUITE 10G

The Oracle Collaboration Suite 10g version 10.1.2 (or OCS 10g) was originally chosen by the TACFIRE research group as the SOA to support the TACFIRE portal [6], [21]. The OCS 10g software suite is separated into two key components: the Oracle Applications Tier and the Oracle Infrastructure Tier [26], [27]. The Oracle Infrastructure Tier is further divided into three parts: (a) an Internet Directory, which houses the Oracle user accounts resident on the OCS, (b) the Oracle Single Sign-on component, which allows Oracle users to access multiple Oracle applications through a single portal, and (c) the Oracle Collaboration Suite database [28]. The Applications Tier contains a total of ten Oracle applications [26]. Table 1 provides a list of the OCS 10g applications, along with a brief summary describing each application's function to the end user.

OCS 10g Application	Function
Oracle Content Services: Oracle Drive rich media client	WebDav (Web-based Distributed Authoring and Versioning) based file and content management rich media client. A downloadable plugin client that allows the user to map a network drive to a specific file directory on the OCS (the 'Oracle Drive'). Files can be dragged and droppped into the Oracle Drive using a mouse.
Oracle Content Services: Web Browser client	Web browser based WebDav (Web-based Distributed Authoring and Versioning) file and content management component. User accesses specific file directory on the OCS using a web browser. Files can be dragged and droppped using a mouse.
Oracle Calendar	Web browser calendar application. Allows user to schedule events centered around other Oracle applications, including (a) RTC Web Conferences, and (b) scheduling Workspace meetings.
Oracle Voicemail & Fax	Private Branch-Exchange (PBX) based voicemail and data facsimile. Works with Oracle Web Mail to store voicemail and fax messages. Requires a Private Branch Exchange to function.
Oracle Workspaces	Web browser based user-definied workspaces. Allows users to leverage other Oracle applications (e.g. schedule meetings, hold discussions, share files, notify via web mail) within a predefined workspace.
Oracle Mobile Collaboration	Mobile email, mobile web meeting, and calendar functions for Smartphones, Personal Data Assistants.
Oracle Discussions	Web browser based discussion boards. Allows users to post responses, post files, start and manage message threads.
Oracle Real-Time Collaboration: Web Conferencing	Web browser real-time collaboration client. Allows users to collaborate real time using a variety of tools. Users chat, use VOIP, draw on a whiteboard, share desktop applications, poll other users, and delegate speaker/presenter roles between other users.
Oracle Real-Time Collaboration: Instant Messager rich media client	Jabber-based rich media chat client. A downloadable plugin client that allows user to chat. Uses Extensible Messaging and Presence Protocol (XMPP) protocol.
Oracle Web Mail	Web browser email client. Stores all messages (including emails, voicemails, and faxes) specific to user in the Oracle Database. Utilizes SMTP and IMAP4 to exchange email with other Oracle users on the same server.

Table 1. Oracle Collaboration Suite 10g Applications [From [26]]

The Oracle Collaboration Suite Installation Guide 10g 10.1.2 provides a variety of deployment options for the OCS software [27], [28]. These options include a large group

of multi-computer configurations and one method for installing the OCS software on a single server (the Single-Computer installation configuration) [27], [28]. The 'multi-computer' configurations deploy the key Oracle components (the Oracle Applications Tier, the Oracle Internet Directory, the Oracle Database, and the Oracle Infrastructure Tier) between two or more computers. For example, one multi-computer configuration includes an option to split the Infrastructure and the Application components between two computers (one computer for the Infrastructure tier components, and one computer for the Application tier components) [27]. In the single-computer installation configuration, the Oracle Database, the Oracle Applications tier, and the Oracle Infrastructure tier all reside on the same computer [27]. The Oracle Installation Guide recommends deploying the software suite on a single-computer for a 'smaller' pool of users (200 to than 1,000 users), and a multi-computer configuration for user groups of over 1,000 [27], [28].

Figure 3 details the location of individual OCS 10g applications in a 'single-computer' OCS configuration [28]. The Oracle applications all reside on the Applications Tier of the OCS (none reside in the Infrastructure Tier) [28]. The way in which Oracle 10g applications connect to the user's terminal can be divided into three distinct categories: (a) through a web browser via an HTTP or HTTPS request, (b) through an Oracle rich media client component (e.g., the Oracle RTC Instant Messenger), or (c) through either a Voice Over IP or a Voice Over XML gateway server, hooked into a Private Branch Exchange (PBX) [28], [29].

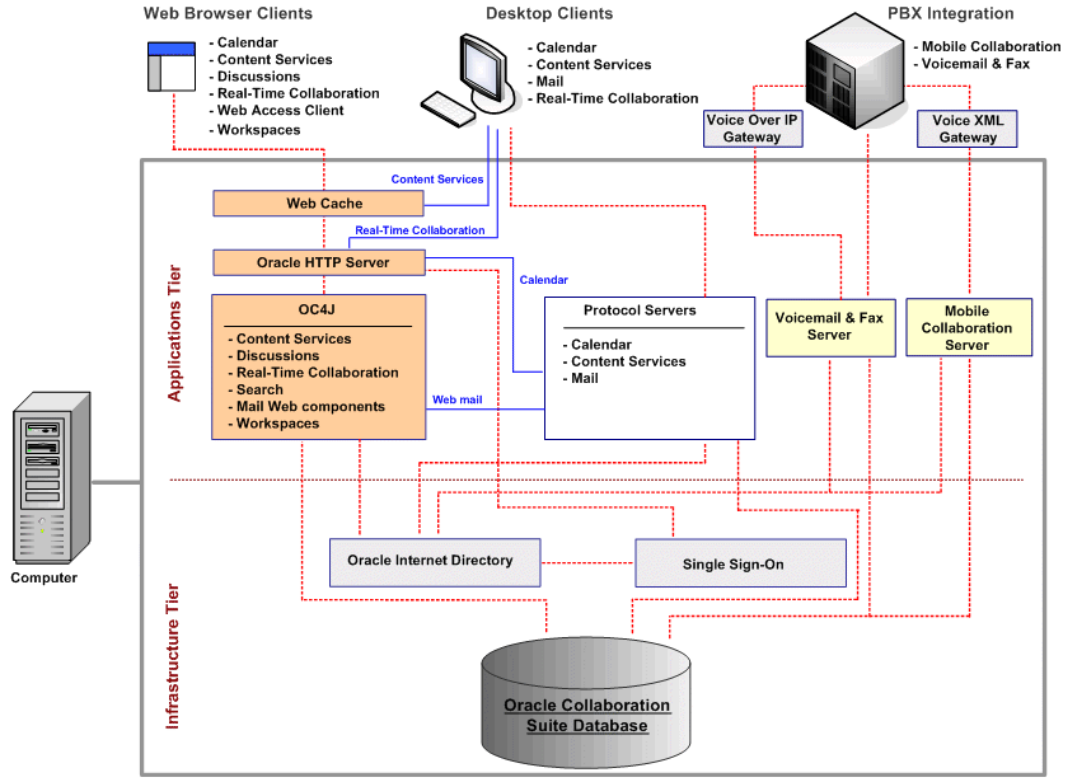

Figure 3. Oracle Collaboration Suite 10g Single Computer Deployment [From [28]]

Nearly all of the applications that connect through a web browser are OC4J applications [30]. 'OC4J' stands for Oracle Container for J2EE (Java Enterprise Environment) [28], [29]. A 'J2EE container' is a Java-based application infrastructure running Enterprise JavaBeans [29]. In OCS 10g, OC4J applications can be managed either as (a) stand-alone components, outside of the Oracle Application Tier infrastructure, or (b) installed and managed as components of the Oracle Application Tier infrastructure [28]. In the case of a OCS 10g single-computer configuration, the OC4J applications are part of the Oracle Application Tier infrastructure [28], [29]. OC4J applications are J2EE 1.3 compliant (Java 2 Enterprise Environment), and run on a standard file-based Java Developer's Kit (JDK) 1.4 Java Virtual Machine [29]. OC4J applications support Java Servlets, Web services, and the following J2EE specific standards: Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), Web Services Definition Language (WSDL), and Universal Description, Discovery and

Integration (UDDI) [29]. By adhering to a handful of generally accepted web standards, OC4J applications provide some degree of interoperability and reuse to the end user (thereby fulfilling two of the basic design characteristics of a SOA) [3], [11], [29].

The OCS 10g applications which are part of the OC4J container include Oracle Content Services, Oracle Discussions, Oracle Real-Time Collaboration Web Conferencing, Oracle Web Mail, and Oracle Workspaces [28]. For these applications, HTTP (or HTTPS) requests are sent from the user to the Oracle Web Cache, which forwards the request on to the Oracle HTTP Server [28]. The Oracle HTTP Server then sends the request to the specific application in the OC4J container [28]. There is one exception to this flow: Oracle Web Mail. The Oracle Web Mail application uses Protocol Servers (e.g., Simple Mail Transport Protocol) to process inbound and outbound email [28].

Although the Oracle Calendar application can be accessed from a web browser, the Calendar application does not reside in the OC4J container [28], [29]. Additionally, the Calendar application handles all HTTP (or HTTPS) Calendar protocol requests through the Calendar Protocol Server, not the Oracle HTTP Server [28], [29].

The rich media clients in the OCS 10g are downloadable, plugin applications for that run independent of a web browser. The OCS rich media clients include clients such as the Real-Time Collaboration (RTC) Instant Messenger and the Content Services WebDav-based ‘Oracle Drive’. The Oracle Real-Time Collaboration Instant Messenger component allows OCS clients to *chat* [30]. RTC Instant Messenger uses Jabber, a form of Extensible Messaging and Presence Protocol (XMPP), to relay chat between clients [30]. The ‘Oracle Drive’ uses WebDav, a set of extensions for the Hyper-Text Transfer Protocol (HTTP). The purpose of these extensions is to allow users to manage and control files and file directories via a remote server [31], [32]. The ‘Oracle Drive’ provides OCS users with a collaborative tool capable for maintaining version control between various file types and directories on the OCS 10g server [32], [33]. It implements the WebDav protocol to provide file synchronization capabilities between the client and Oracle Content Services on the OCS 10g server [32], [33]. The associated file directory can also be accessed from via a browser on the client. Since WebDav

functionality had been extensively demonstrated and tested as a multilevel-aware application on an XTS-400, the OCS 'Oracle Drive' component seemed like the ideal candidate to include in the testing [31]. More details regarding these two components are described in Table 1. These applications are not part of the OC4J container. Some of the clients (e.g., RTC Instant Messenger) are legacy Oracle applications that were adopted from previous Oracle enterprise platforms for use in OCS 10g [34].

Regardless of location or type, all of the individual Oracle applications retain the ability enrich the user's collaboration environment by reaching out and utilizing the *other* Oracle applications within the OCS software suite. As described in the Oracle Collaboration Suite 10g Concepts Guide, the Oracle applications (Oracle Mail, Oracle Calendar, Real Time Collaboration (Web Conferencing and Instant Messaging), Content Services, Workspaces, and Discussions) can utilize each other on an ad-hoc basis, based on the user's needs [35]. For instance, the Oracle Workspace application allows users to share documents (Content Services), collaborate with team members (Web Conference), hold discussions (Oracle Discussions), and manage tasks [32], [35]. From the Workspace Application, a user can schedule a web conference for the other workspace members, announce the meeting via email notification, or post discussion threads specific to the workspace [35]. This on-demand pull allows the Workspace Application to access the other Oracle applications on an ad-hoc basis. To simplify and limit the scope of such 'cross-application' testing, the Oracle Workspace application was the only application tested in this manner.

The next evolution of the Oracle Application Server (Oracle Fusion Middleware) is planned to unify the Oracle applications under a single set of 'web-service' based standards [29], [34], [36], [37]. *All* of the applications in Oracle Fusion Middleware will be built specifically for use in OC4J containers, and therefore will also adhere to the J2EE standards specific to SOA: XML, SOAP, WSDL, and UDDI [29]. Unfortunately, such 'uniformity' requires an immense amount of labor to bring the non-OC4J applications up to spec [34], [36]. This unification, while costly, does provide additional web service advantages [3], [14], [34], [38]. The 'uniformity' of applications provides users with web service tools similar to those provided in Web 2.0 technology: (a) rich

internet applications ("getting the web browser to act as a desktop"), (b) collaboration tools (e.g., wikis, blogs, real-time collaboration enablers), (c) user-contributed content (large scale web environments in which users share content), and (d) using the web itself as a platform ("…where the internet is a data source and a platform for services") [38]. SOA users can build their own content-rich, web services and applications, including discussion boards, links, announcement pages, and Wiki portals [14], [36], [38]. XML based Friend-of-a-friend (FOAF) files can be used to build bulletin boards that are 'semantically enabled:' if one user puts a specific word (or string of words) into his or her content, it will link up to other users who include the same word in their content [14], [34], [38].

E. MYSEA

The Monterey Security Architecture (MYSEA) project of the Naval Postgraduate School seeks to demonstrate how participants from various U.S. agencies can access different levels of classification (without violating classic confidentiality policy modeled by Bell and LaPadula [9]) via a single, commercial-off-the-shelf (COTS) based client. The largest need to achieve this configuration comes from the growing need to share information and intelligence across networks with different classification levels (Unclassified, Secret, Top Secret), and between coalition partners and allies (NATO, OIF, OEF). Two of the guiding purposes of the MYSEA project are (1) "support research in high assurance multilevel security (MLS)," and (2) to "support research in dynamic security." Developments regarding the first purpose (high assurance multilevel security) have focused on utilizing high assurance servers (e.g., an XTS-400) to enforce a unified mandatory access control policy over untrusted COTS client hardware [7], [8], [31].

Included in the MYSEA project is a working multilevel secure testbed. The testbed contains no operational data an all confidentiality levels are simulated. It employs stateless commercial-off-the-shelf (COTS) hardware and software-based clients. Users log in at clients and select the session level at which they wish to work, bounded, of course by user authorizations, such as clearance. Thus at any one time, different clients support sessions at different sensitivity levels. Figure 4 is an illustration of the MLS

controlled environment present in the MYSEA lab. The security policy is enforced by high assurance servers (XTS-400 servers) and Trusted Path Extension devices. In this setup, the stateless clients can safely traverse multiple sensitivity levels without fear of compromising classic confidentiality policy, as modeled by Bell and LaPadula. Several application servers have been tested in this controlled environment, including C2PC, various mail servers, and an Apache web server. Several protocols have been on MLS high assurance servers as 'MLS Aware' applications. The applications have been modified to allow them to reside and function as single level applications on the MLS server itself. As a result, an 'MLS Aware' application is able to read down to appropriate information at lower security levels. SMTP, IMAP, and WebDAV are some of the protocols that have been adapted to be as 'MLS Aware' in the MYSEA testbed [7], [8].

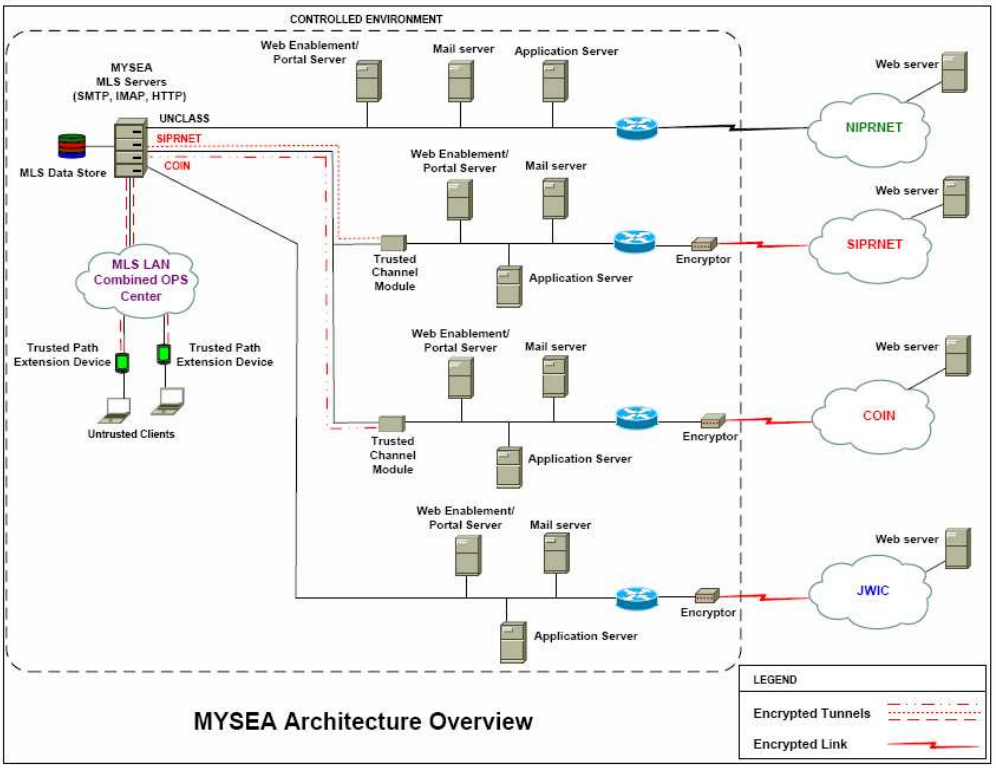

Fig. 1. Distributed Multilevel Secure Architecture.

Figure 4. Distributed Multilevel Secure Architecture [From [7]]

THIS PAGE INTENTIONALLY LEFT BLANK

III. DESIGN

This chapter describes a high level design used to incorporate a web based Service-Oriented Architecture (SOA) into a multilevel environment. The design served as a proof of concept for the integration of a SOA into a multilevel secure environment. The first section of this chapter will describe the goals associated with integrating a web-based SOA software suite into the simulated MYSEA multilevel testbed. The second section elaborates on the actual high-level design utilized. The third section summarizes the first three sections of this chapter.

A. GOAL

The goal of this thesis is to determine the feasibility, or proof of concept, of incorporating a web-based SOA software suite into a multilevel environment. As discussed in Chapter 2, the incorporation of a Service Oriented Architecture (SOA) software suite into a multilevel secure environment has neither been tested nor implemented. Exhaustive application testing on the TACFIRE portal verified that SOA-based architecture and SOA web services were compatible with existing U.S. Navy legacy networks. However, cross-domain testing has not been conducted on the TACFIRE portal [6]. This project intends to combine the benefits associated with a web-based SOA with the security capabilities of a multilevel environment [8], [21], [26]. The MLS ability to 'read down' could provide a SOA-based file content service with the ability to pool similar intelligence from different sensitivity levels. Intelligence aggregation could be improved tremendously in this manner, as mission critical data from lower sensitivity levels could be instantly aggregated with higher level data via a single, web-based multilevel aware SOA search application. Implementing a SOA server into each of the various DoD networks (NIPRNet, SIPRNet, CENTRIX, etc) would provide the benefits of the SOA-based services into each network individually. In this configuration, users could utilize SOA-based real-time collaboration applications to chat or hold web conferences on any network classification they have access to, from a single,

stateless COTS-based client. A further requirement of this experiment is to specify which, if any, web-based services of a SOA software suite function in a multilevel environment.

B. DESIGN

Because of the experimental nature of this work, a copy of a segment of the testbed was created for use in this project, so that normal MYSEA testbed operations would not be disturbed. The segment of the MYSEA testbed copied was the simulated SIPRNet enclave. In the MYSEA testbed, this segment includes various application servers, including C2PC, a Linux mail server, and an Apache web server. The existence of these additional application servers made the simulated SIPRNet enclave the most desirable segment to copy. **Note:** In terms of application servers, the segment copy used for this research only included a single Linux mail server. The segment copy did not include replicas of the other application servers (e.g., C2PC, Apache Web Server) in the MYSEA testbed. The segment copy used in the experiment is referred to as the 'simulated multilevel environment' in this and chapters to follow.

Since this project focuses only on a qualitative 'yes or no' regarding the functionality of a SOA service, system performance of the machine(s) utilized will not be a concern. Network load testing will also not be a consideration, since this project's primary goal is to simply verify the basic functionality of the services of a SOA software suite.

In terms of network topology used to test the SOA software suite, the final architecture used as a proof of concept will only include (a) the SOA software suite, hosted on one or more servers, (b) a single multilevel secure server acting as proxy, (c) two clients, and (d) a separate SMTP mail server to test the SOA software suite's ability to exchange mail via the SMTP protocol. True multilevel testing was not considered for this experiment. Although testing a SOA software suite across multiple levels should be considered for future research, the current experiment is limited to single level testing only. This experiment sought to limit variation in the test results as much as possible, and by simply testing across a single level, disparity between the results could be isolated to

either (a) the SOA software suite employed, or (b) the multilevel server acting as the proxy. The clients will each authenticate with the multilevel server via a Java-based virtual trusted path extension (TPE) device known as the TCBE (Trusted Computing Base Extension), and then attempt to access the SOA software suite's services. Figure 5 illustrates the network topology (and associated connections) required by the high-level design of this project. Both of the clients will establish a single level Hyper-Text Transfer Protocol (HTTP) connection with the OCS Server (e.g., the server hosting the SOA software suite) using the multilevel secure (MLS) server as a proxy. For each client, the single level connection will be through a virtual trusted path extension device (the TCBE), which provides an extension of the trusted path between the client and the multilevel secure server [7], [8]. A Linux Mail Server, configured to exchange email via SMTP, will be connected to both the OCS Server and the multilevel secure server.

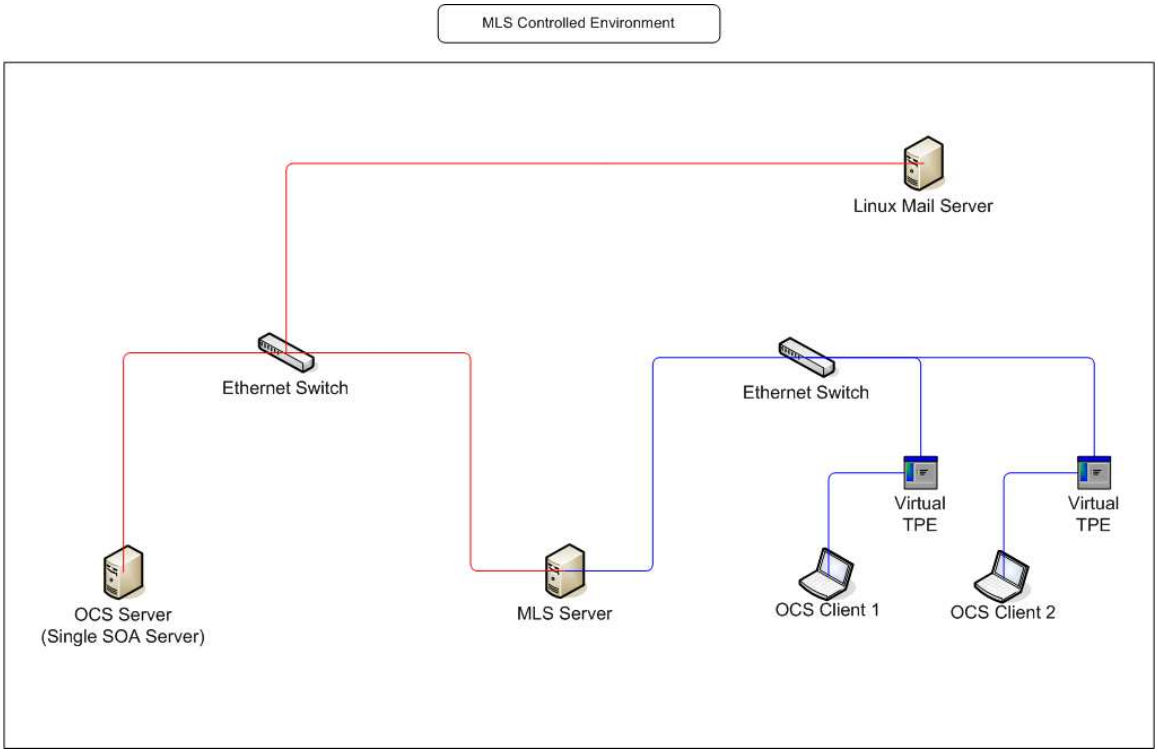

Figure 5. Network Topology of High Level Design.

The TACFIRE research project proved that a web-based SOA software suite could be successfully integrated into the existing Navy legacy network infrastructure with minimal configuration changes. Web-based SOA services provide a secure, scalable,

user-initiated means of hosting applications to geographically dispersed clients. As discussed in Chapter II, this type of service is ideal for use in the Navy's networks. Focusing on web-based services will reduce the number of TCP/IP ports involved in the experiment, thereby simplifying the analysis. The majority of the services (or all, if possible) provided by the SOA software suite in this experiment should be web-based services. TACFIRE servers have been deployed onto both Secret Internet Protocol Router Network (SIPRNet) and Unclassified but Sensitive Internet Protocol Router Network (NIPRNet) DoD legacy networks [6]. Since the two networks (NIPRNet and SIPRNet) are air-gapped, there is no interaction between a TACFIRE server on the NIPRNet, and a TACFIRE server deployed on the SIPRNet. The working prototype for the TACFIRE portal utilized Hyper-Text Transfer Protocol (HTTP) with Secure Socket Layer Encryption (SSL), or HTTPS, to meet Defense Information Systems Agency (DISA) security requirements for web based applications. [1], [6], [15]. For this reason, HTTPS should be included as a configuration option in the SOA software suite used in this experiment. Although HTTPS will be included as a configuration option on the SOA software suite used, HTTP Hyper-Text Transfer Protocol (HTTP)) will be used instead. Randy Maule iterated that this experiment should be a simple proof of concept involving the least complicated port configurations available (in this case, Port 80, or HTTP). Additionally, Professor Maule indicated once the SOA software suite was configured for HTTP, the transfer of services on a SOA software suite over to HTTPS would be both nontrivial and inconsequential to the outcome of this experiment.

Similar to the TACFIRE portal, the high level design used in this experiment is by no means a 'pure SOA.' The services tested will be those resident on the SOA software suite. To meet the definition of a pure SOA, all of the existing applications (and associated application servers) would have to be redesigned and incorporated into the SOA software suite itself. In this design, no external application servers or applications residing on the MYSEA domain will be modified. Considering both the experimental nature of the testbed and the monstrous amount of labor required to transfer the existing MLS applications into the SOA software suite, the 'pure SOA' concept was not implemented [7], [8].

While the wholesale conversion and introduction of external applications into a SOA was omitted from the scope of this experiment, the possibility of SOA applications *interacting* with identical external application servers was not. SOA software suites usually include the functionality to interact with common Internet Protocol based protocols like Simple Mail Transfer Protocol (SMTP). For example, in the Oracle Collaboration Suite 10g, the administrator can configure the OCS to interact with an external SMTP mail server by changing the OCS' SMTP settings via the web-based Oracle Application Control Console [16], [39]. This type of functionality is particularly advantageous in the MYSEA environment, since several IP-based protocols (including SMTP) have been tested on application servers in the multilevel testbed [8], [31]. Part of this experiment will include testing to determine if a SOA software suite can exchange information with an existing MYSEA simulated SIPRNet enclave application server across the same IP-based protocol.

The design characteristics discussed in this section provided a sufficient list of top level requirements for the SOA-based software suite to be utilized in this project. The SOA software suite, network topology, and other components must meet the following requirements:

- The hardware and software used in this experiment will include (a) the SOA software suite, hosted on one or more servers, (b) a single multilevel secure server acting as proxy, (c) two clients, and (d) a separate SMTP mail server to test the SOA software suite's ability to exchange mail via the SMTP protocol.
- The actual MYSEA testbed will not be used in this experiment.
- A copy of a segment of the testbed will be created for use in this project.
- This copied segment will represent the simulated SIPRNet enclave in the MYSEA testbed.
- The majority of the SOA software suite's services (more than half) will be web based; the SOA software suite's web services will support both HTTP and HTTPS.

- The SOA software suite's web based-services will utilize HTTP in this experiment.
- No external application servers or applications residing on the MYSEA domain will be modified
- The SOA software suite will include the functionality to exchange mail with another external mail server via Simple Mail Transfer Protocol (SMTP).
- System performance and network load testing will not be conducted in this experiment.
- Testing in the simulated multilevel environment will be Single Level only.

C. SUMMARY

The design specifications detailed in the first two sections of this chapter provide a high level specification for incorporating a SOA software suite into a multilevel environment. A prototype based on the top level requirements in this chapter was implemented for this project, and is described in Chapter IV.

IV. IMPLEMENTATION

This chapter will describe how the Oracle Collaboration Suite 10g (OCS 10g) was integrated into the simulated multilevel testbed. The first section of this chapter describes the SOA software suite that was chosen for this experiment, and how it was deployed. The second section discusses the OCS services that were deployed in the simulated multilevel testbed. The third section discusses the OCS services that were not deployed into the multilevel testbed, along with a short explanation detailing why. The fourth section is a summary of this chapter.

A. SELECTION OF SOA SOFTWARE SUITE FOR THIS PROJECT

For this project, the Oracle Collaboration Suite 10g, version 10.1.2, was selected as the SOA software suite to introduce into a multilevel environment. The Oracle Collaboration Suite, or OCS, met all of the top level requirements described in Section B of Chapter III. As discussed in Chapter II, the OCS 10g was chosen as the SOA software suite for test and evaluation in the TACFIRE research portal. OCS 10g provides a robust array of web services, the majority of which are accessible through a web browser [28]. Deployed users can collaborate and exchange information real-time by simply accessing the OCS services via a browser. These characteristics made OCS 10g an ideal candidate for the TACFIRE portal. The TACFIRE portal, using the OCS 10g software suite as its test model, had been successfully deployed and tested in TRIDENT WARRIOR 05, TRIDENT WARRIOR 06, and TRIDENT WARRIOR 07 fleet exercises [6], [21], [22]. TRIDENT WARRIOR 05 testing proved that remote OCS 10g servers located stateside could provide services to client workstations on ships underway (operating from host-based services on ships) [6]. OCS 10g's success as a TACFIRE SOA prototype provides justification for its selection as the SOA software suite to use in this project.

B. OCS APPLICATIONS DEPLOYED

The following applications of the Oracle Collaboration Suite (OCS) 10g, version 10.1.2, were deployed and tested in the Monterey Security Architecture (MYSEA) multilevel testbed:

- Oracle Web Mail (web browser)
- Oracle Content Services: "Oracle Drive" (rich media client)
- Oracle Content Services (web browser)
- Oracle Calendar (web browser)
- Oracle Workspaces (web browser)
- Oracle Discussions (web browser)
- Oracle Real-Time Collaboration (RTC) Instant Messenger (rich media client)
- Oracle Real-Time Collaboration: Web Conferencing (web browser)

A description of each of these Oracle applications is presented on Table 1, located in Chapter II. The items to the left of the applications listed in parenthesis (e.g., web browser and rich media client) are the tools used to access the associated OCS 10g service.

To simplify the testing process, all of the OCS applications deployed (with the exception of the Oracle Real Time Collaboration Instant Messenger component and the Oracle Content Services "Oracle Drive") were web-based applications that connected to the OCS via the OCS 10g HTTP Server (Port 80) [28], [40]. This reduced the number of client applications required for testing to (a) a single web browser (either Internet Explorer 7.0 or Mozilla FireFox), and (b) the two downloadable plugin applications, RTC Instant Messenger and the Content Services "Oracle Drive". Hypertext Transfer Protocol over Secure Socket Layer, or HTTPS (Port 443), was not configured on the OCS server due to the complexity of the configuration task [40]. Configuration of an OCS server for HTTPS is non-trivial and must be initiated at the beginning of the OCS software installation. It has been deferred for future work.

In addition to testing the services listed above, this experiment included a simple 'email exchange' test between the OCS Sever and an external mail server using the Simple Mail Transfer Protocol (SMTP) [41]. The OCS software suite has the functionality to exchange email with another, external mail server. The `SMTP In` and `SMTP Out` port settings in the Protocol Server on the OCS Applications Tier (see Figure 3 in Chapter II for a layout of the Application Tier components) are reconfigured to match those of the external mail server [39]. The settings associated with these ports can be changed using the Oracle Application Control Console, a browser-based configuration tool resident on the OCS Server [27], [39]. The simulated SIPRNet enclave of the MYSEA environment included a Linux operating system-based SMTP server that had been successfully implemented and tested on the simulated multilevel testbed for access via the XTS-400 [8]. For the OCS Server to exchange email via SMTP with this pre-existing Linux server, only the OCS Server would need to be configured. The Linux based SMTP server would require no changes. Therefore, errors encountered in SMTP testing would be due to problems at either the OCS Server or the proxy server (the XTS-400).

C. OCS APPLICATIONS NOT DEPLOYED

The following applications of the Oracle Collaboration Suite (OCS) 10g, version 10.1.2, were *neither* deployed *nor* tested in the Monterey Security Architecture (MYSEA) multilevel testbed:

- Oracle Voicemail and Fax
- Oracle Mobile Collaboration

The Oracle Voicemail and Fax application of the OCS 10g required the existence of a Private Branch Exchange (PBX) to function [27], [33]. Since the MYSEA multilevel testbed did not have a PBX, this application was not implemented. Also, since the Voicemail and Fax OCS application is not a web-based application, even had a PBX existed in the MYSEA multilevel testbed, this application would be beyond this project's scope.

The Oracle Mobile Collaboration application of the OCS 10g provides a variety of mobile services for mobile phones, Blackberrys, Smartphones and Personal Digital Assistants (PDAs), including Mobile Push Mail and Mobile Collaboration [26], [27]. Since the MYSEA simulated multilevel testbed provides no interface for any of these devices, this application was not implemented.

D. SUMMARY

The implementation discussed in the chapter supports a set of applications sufficient to demonstrate the proof of concept pertinent to this project: the integration of a SOA into a multilevel secure environment. The SOA software suite and specific applications selected for use in this project meets the high level requirements outlined in Chapter 3. Also, the SOA software suite utilized, the OCS 10g, was chosen because it has been successfully implemented and tested for use as a SOA web service portal in the fleet [6], [21]. The next chapter details the functional test plan based on this implementation and the test results.

V. TESTING AND RESULTS

This chapter describes both the test design and the test results obtained in this experiment. As discussed in Chapter IV, this experiment seeks to conduct proof-of-concept testing on the services of a Service-Oriented Architecture (SOA) software suite integrated into a multilevel secure environment. The first section of this chapter discusses the functional test plan that was developed for this project. The second section describes the test results observed when the SOA server was directly connected to the clients via a single switch. The third section details the test results observed when the clients were connected to the SOA server via a multilevel proxy server (an XTS-400).

A. FUNCTIONAL TEST PLAN OVERVIEW

This section includes the functional test plan utilized for this experiment. Testing the OCS 10g software was accomplished in two stages. The first stage sought to establish a baseline of 'expected results' for selected OCS applications. The second phase of testing determined if selected OCS services could function in the simulated multilevel environment (across a single level), and if the selected services were interoperable with existing identical application servers (ex. Simple Mail Transfer Protocol) residing on the simulated SIPR enclave of the MYSEA multilevel testbed. All experimentation was conducted in Glascow EAST, Room B04, at the Naval Postgraduate School.

The OCS applications were tested individually for their own specific functionality. As detailed in Appendix B, each application was tested using the simplest testing required to verify its functionality. It should be noted that the individual Oracle applications retain the ability enrich the user's collaboration environment by reaching out and utilizing the other Oracle applications within the OCS software suite. To simplify and limit the scope of such 'cross-application' testing, the Oracle Workspace application was the only application tested.

The procedures to install these components are outlined in Appendix A, Installation Procedures. Prior to both stages of testing, all of the hardware listed in Parts 1 through 4 of this Section was assembled and connected.

B. COMPONENTS USED IN TESTING

This section describes the components used in testing. The first part explains how the OCS software was deployed. The second part discusses how the clients used in this experiment were configured. The third and fourth parts briefly describe the XTS-400 server and the Linux mail server utilized in the second stage of testing, respectively.

1. Deployment of the OCS Software Suite

The Oracle Collaboration Suite 10g Release 1 (10.1.2) was installed on a single computer running Windows Server 2003 (Service Pack 2 installed). Although the OCS 10g software could have been installed on a computer running a Linux-based operating system, Windows was chosen based on the installer's previous experience with Windows operating systems. This platform was a Dell Dimension 4600 with a Pentium IV 3.0 Ghz processor, 2 GB of RAM, 20 GB of available hard drive space, an input jack for a microphone, an input jack for headphones, and a Network Interface Card to support an Ethernet connection. Additionally, two browsers were installed on the OCS Server: Internet Explorer 7.0 with the Java Runtime Environment Version 6, Update 3 plugin, and Mozilla FireFox 2.0.0.11. It met the minimum hardware specifications to run the Oracle Collaboration Suite on a machine running Windows Server 2003, as described in the Oracle Collaboration Suite Installation Guide 10g 10.1.2 [27]. This computer, labeled 'OCS Server' in testing, was also setup as a standalone terminal server. No server roles were configured in Windows Server 2003. The Oracle Namespace in the Oracle Internet Directory is specified as `cisrlabmlstestbed3.com`

2. OCS Clients

The clients used in this experiment were referred to as 'OCS Client 1' and 'OCS Client 2.' Both were Dell laptop computers running Windows XP (Service Pack 2

installed); each had a Pentium III 1.0 Ghz processor, 1 GB of RAM, 10 GB of available hard drive space, an input jack for a microphone, an input jack for headphones, and a Network Interface Card to support and Ethernet connection. Both OCS Clients had the same web browsers installed as the OCS server: Internet Explorer 7.0 with the Java Runtime Environment Version 6, Update 3 plugin, and Mozilla FireFox 2.0.0.11. To support a single level connection across the multilevel testbed, both clients also had the Virtual Trusted Path Extension Device executable file (`tcbe.exe`) installed on the Windows desktop. This component, also known as the 'Virtual TPE,' is a software application that exhibits the same functionality as a hardware Trusted Path Extension (TPE). As a functional prototype, it does not actually provide a true extension of the trusted path between the XTS-400 server and the untrusted client (in this experiment, OCS Client 1 and OCS Client 2). After starting the Virtual TPE application, the user presses the SAK, establishes communication with the MYSEA server, and then selects a session level. For the purposes of this experiment, the user selects the session level corresponding to the simulated SIPRNet enclave: SIM_SECRET. From that point on, the user's requests from the untrusted client will be recognized by the XTS-400 as those of the session level selected on the Virtual TPE [7], [8].

3. XTS-400 Server

An XTS-400 server running STOP 6.1 operating system was configured to serve as the proxy server for the OCS Clients to access the services on the OCS Server [8]. The XTS-400 included two Network Interface cards. The OCS Clients used the XTS-400 as a proxy server, sending requests to the OCS Server via the XTS. Note: the XTS was configured to only allow Port 80 and Port 25 requests from the OCS Clients to reach the OCS Server via Virtual TPE single level connections. Other IP Ports (e.g., Port 443) were not configured for use in this experiment. The XTS-400 was configured by the CISR staff.

4. Linux Mail Server

The simulated SIPRNet enclave of the MYSEA testbed includes a Linux mail application server. This mail server has been successfully tested on SIM_SECRET as an external mail server using the Simple Mail Transfer Protocol (SMTP). Because of (a) the existence of a working SMTP mail server on the MYSEA testbed, and (b) the ease of reconfiguring the OCS Server to exchange mail with a Linux server, this protocol was ideal for testing.

In this experiment, the OCS Server and the Linux Mail server were tested to see if they could exchange email via Simple Mail Transfer Protocol (SMTP). The `SMTP Inbound` and `SMTP Outbound` settings on the OCS Server Application Control Console were modified to communicate with the Linux Mail server on the simulated SIPR enclave. No configuration changes were made to the existing Linux Mail server residing on the simulated SIPR enclave. This test was the first attempt to connect an OCS application service (the OCS' Simple Mail Transfer Protocol) up to an existing, identical application server (SMTP mail server) residing on the simulated SIPR enclave of the multilevel testbed.

C. TESTS

This section describes the two phases of testing used in this experiment. The first section discusses the first phase of testing, where the OCS applications were tested with the OCS Server directly connected to the OCS Clients. The second section describes the second phase of testing, where the OCS applications were tested at the single level using the XTS-400 as a proxy.

1. Network Topology for Direct Connection Testing

In the first phase of testing, the OCS Server was configured to operate in an intranet isolated from the simulated MYSEA multilevel testbed. This initial stage required an OCS server (labeled 'OCS Server') to be connected directly to two clients (labeled 'OCS Client 1' and 'OCS Client 2') via a single switch. The OCS server was

initially configured with an IP address and domain name specific to the simulated SIPR enclave of the multilevel testbed (`cisrlabmlstestbed3.com`). The results of this phase were intended to provide a baseline of what to expect from a correctly functioning OCS service in future testing. Neither the XTS-400 server nor the Linux mail server were utilized in this phase. The hardware deployment for this configuration is depicted in Figure 6.

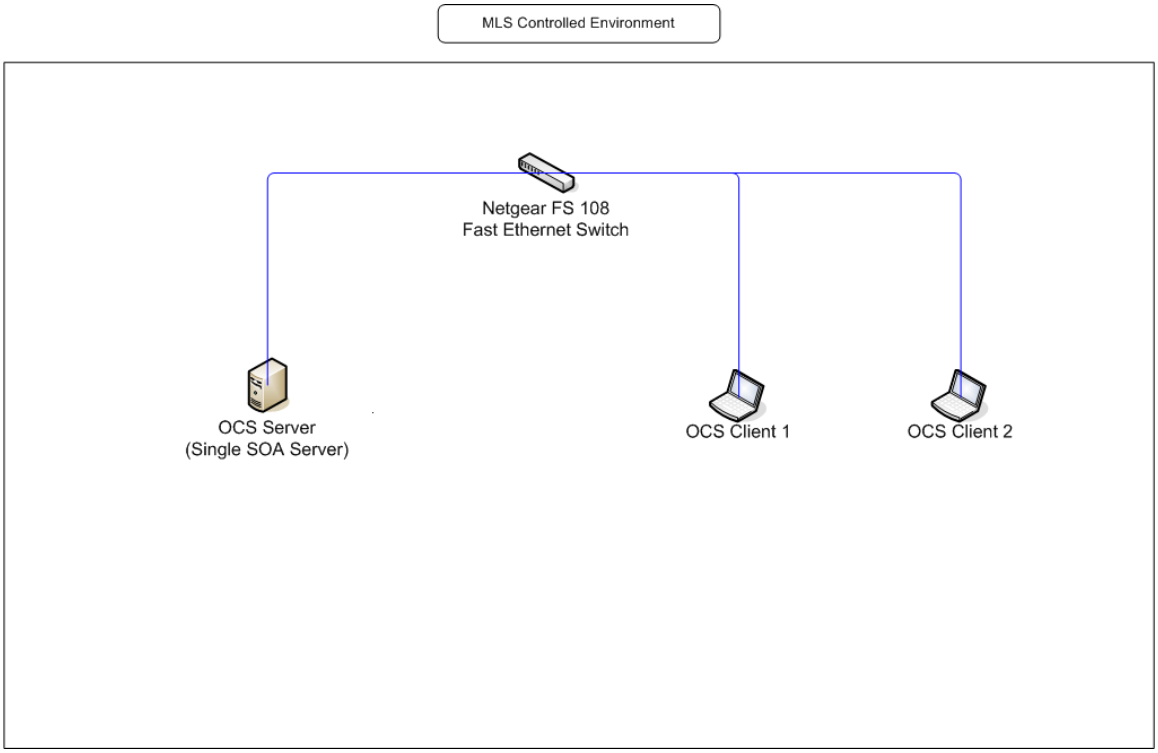

Figure 6. Network Topology for Direct Connection Testing

As outlined in Appendix B, the following applications of the Oracle Collaboration Suite (OCS) 10g, version 10.1.2, were tested while the OCS Server was directly connected to OCS Clients:

- Oracle Web Mail (web browser)
- Oracle Content Services (web browser)
- Oracle Content Services: 'Oracle Drive' (rich media client)
- Oracle Calendar (web browser)
- Oracle Workspaces (web browser)

- Oracle Discussions (web browser)
- Oracle Real-Time Collaboration: RTC Instant Messenger (rich media client)
- Oracle Real-Time Collaboration: Web Conferencing (web browser)

The results of these tests are discussed in Section D of this Chapter.

2. Network Topology for Testing the OCS Services at the Single Level using an XTS-400 as a Proxy

In the second phase of testing, an XTS-400 server was incorporated to serve as a proxy server between the clients and the OCS server. The second phase determined if select OCS services could function in a simulated multilevel environment, although at a single level. Once a connection was established between the OCS Clients and the OCS Server, the selected OCS services were tested. Results of this testing were compared to the 'expected (service) results' garnered during the first phase of testing.

Additionally, the XTS-400 server was connected to a Linux Mail server residing on the simulated SIPR enclave. Further discussion regarding simulated multilevel testing with this server is detailed in Part 4 of Section A in this chapter. Figure 7 details the network topology employed for this phase of testing.

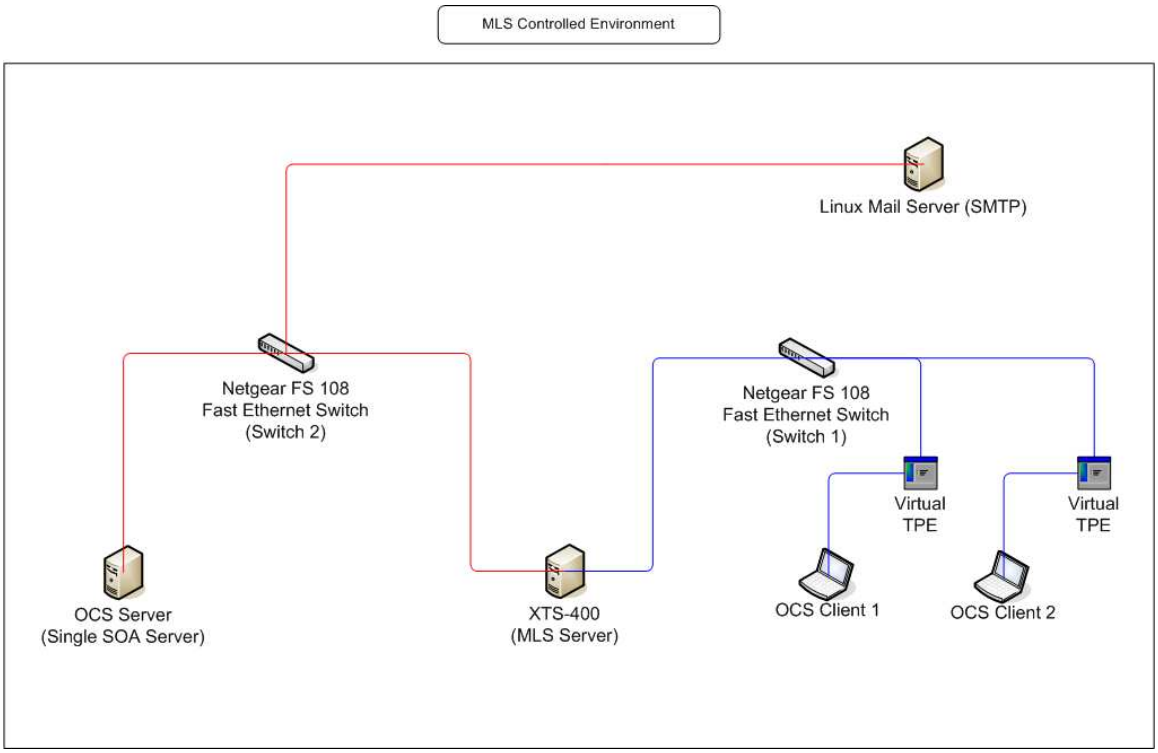

Figure 7. Network Topology for Simulated Multilevel Testing

As outlined in Appendix B, the following applications of the Oracle Collaboration Suite (OCS) 10g, version 10.1.2, were tested at a single level in this experiment:

- Oracle Web Mail (web browser)
- Oracle Content Services: 'Oracle Drive' (rich media client)
- Oracle Content Services (web browser)
- Oracle Calendar (web browser)
- Oracle Workspaces (web browser)
- Oracle Discussions (web browser)
- Oracle Real-Time Collaboration: RTC Instant Messenger (rich media client)
- Oracle Real-Time Collaboration: Web Conferencing (web browser)
- SMTP mail exchange (with Linux server)

The results of these tests are discussed in Section D of this chapter.

D. TEST RESULTS

This section details the results of the experiments. The functional tests performed with the OCS Server directly connected to the OCS Clients established a baseline of 'expected results.' If a particular OCS application did not work in this ('Direct Connection') implementation, then it was not expected to work with at the single level with the XTS-400 serving as a proxy. Following completion of the Direct Connection testing, the network topology was rearranged (as shown in Figure 5) and the applications were tested again. With the exception of the RTC Instant Messenger and Content Services 'Oracle Drive' rich media clients, all of the other Oracle applications were tested via web browser links on the Oracle Collaboration Suite Portal page, as illustrated in Figure 8. (Note: This page is referenced by two different names in the Oracle Documentation. The Collaboration Suite Portal page is also known as the Single Sign-On (SSO) Page in the Oracle Collaboration Suite Administrator's Guide 10g Release 1 (10.1.2) [26].) A detailed description of the test procedures is available in Appendix B. Analysis of the test results will be provided in Chapter VI.

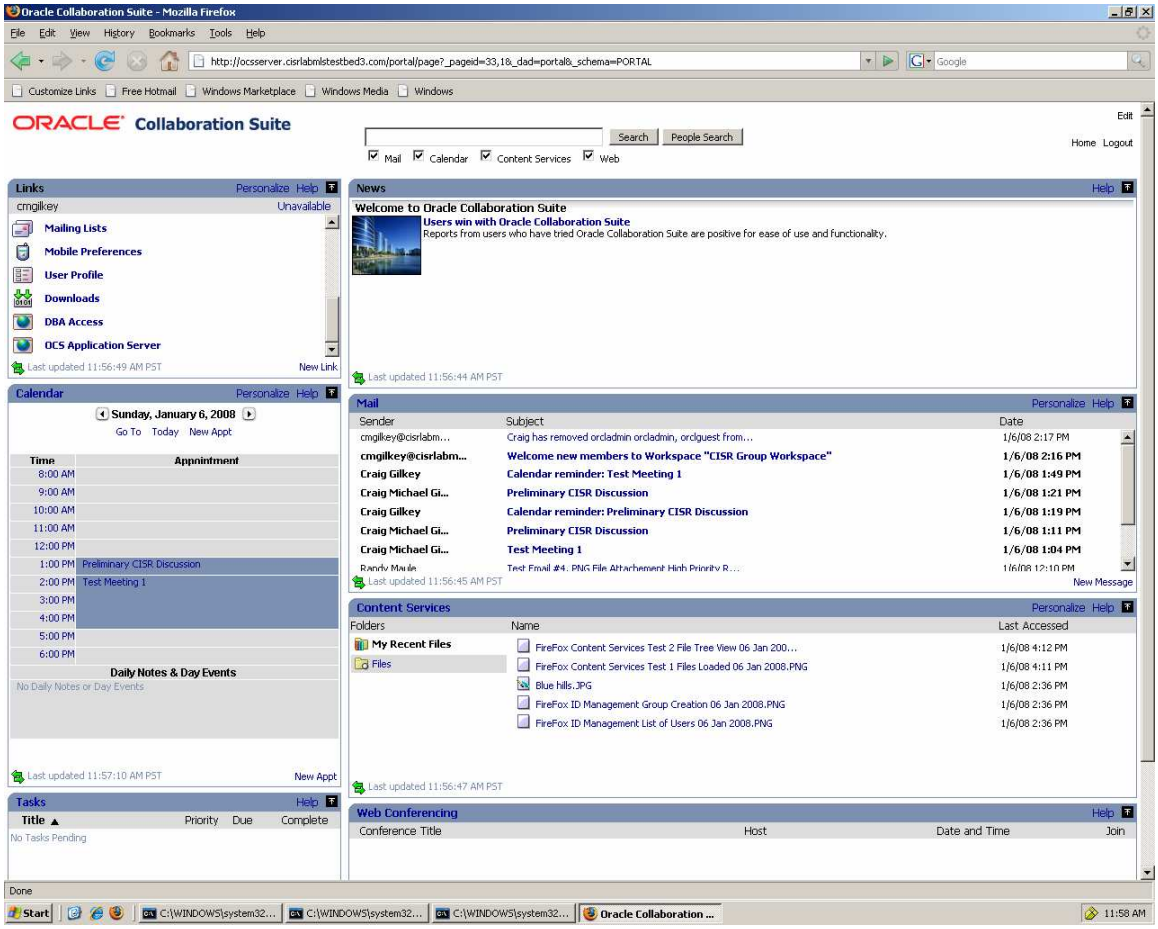

Figure 8. Oracle Collaboration Suite Portal with links to Oracle applications.

1. Oracle Web Mail (web browser)

The Oracle Mail application performed as expected on both OCS Clients directly connected to the OCS Server. The Oracle Web Mail application could be accessed via a web browser, and successfully sent an email to another Oracle User Account (as described in Appendix B). Both the FireFox browser, or the Internet Explorer browser on the clients worked successfully. Multiple Oracle users sent, received, replied, and forwarded both simple text emails (‘The Quick Brown Fox Jumps Over the Lazy Dog’), and emails with attachments (a Portable Networks Graphic [PNG] image file showing a screen capture of the OCS Server’s home portal) [26], [39]. Searching the Oracle Corporate Directory inside the Oracle Mail application for known users was also successful.

Single level testing of the Oracle Mail Application with the XTS-400 serving as a proxy yielded results nearly identical to those observed in direct connection testing. However, one peculiar error was noted during testing in this configuration. An anomaly was observed during the generation of new emails in the Oracle Mail application. When a new email was generated, the text box of the Oracle Mail browser window included HTML source code. Figure 9 is a screen capture of this error. Nonetheless, messages constructed in this manner were transmitted and received correctly; the received messages (that were generated with this error) did not display the HTML code, and the text received in the email mirrored the text sent [32]. The Oracle Mail Administrator's guide revealed that the format of such messages could be changed from 'HTML' to 'Plain Text' using the 'Format' toolbar in the Oracle Web Mail application [39]. It was noted that erroneous email messages had been generated with the 'HTML' format selected When the format of the Oracle Web Mail messages was changed from 'HTML' to 'Plain Text,' the anomaly ceased to exist: all of the 'Plain Text' messages created did not include the HTML source code.

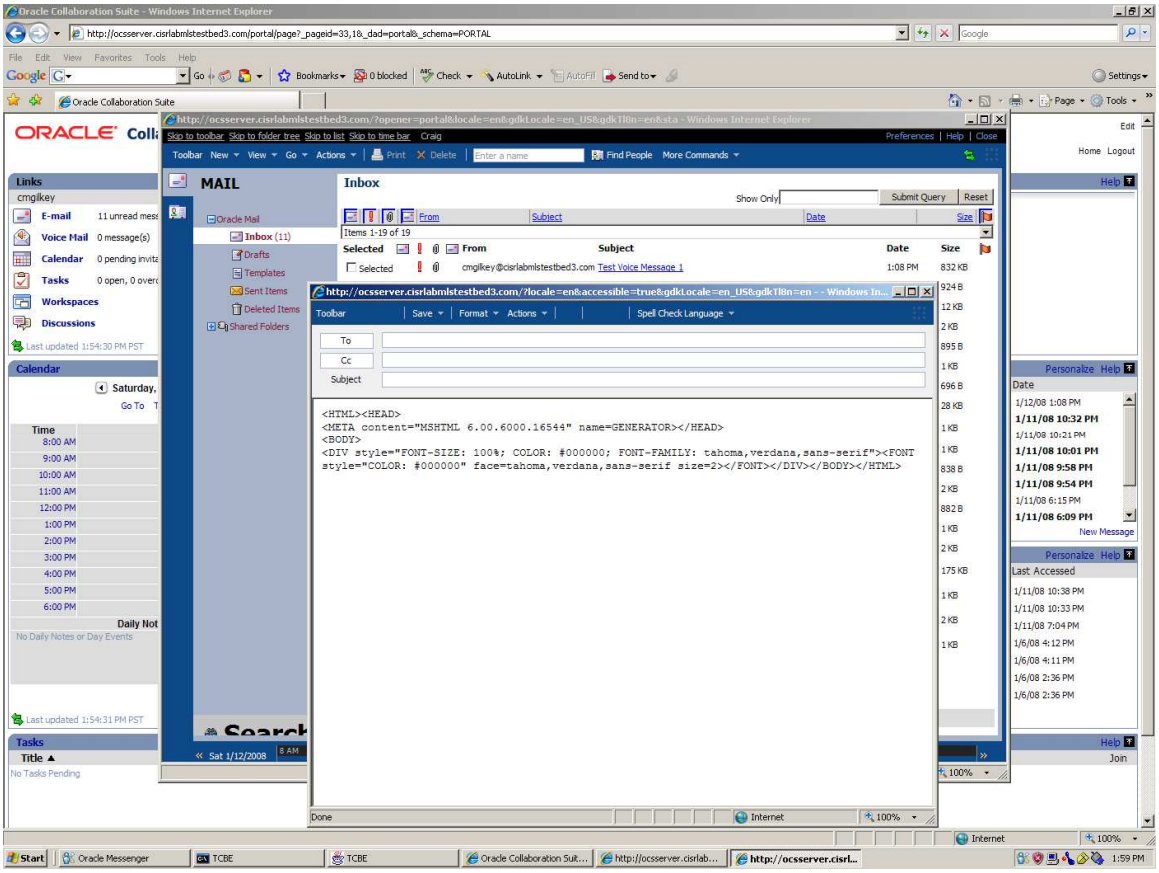

Figure 9. Oracle Collaboration Suite Mail Application with HTML code in the text of a new email message.

2. Oracle Content Services (web browser)

The Oracle Content Services application performed as expected on both OCS Clients directly connected to the OCS Server. The Oracle Content Services application could be accessed via a web browser, and a file was successfully uploaded from the client's desktop file directory into the Oracle User folder (as described in Appendix B). Both the FireFox browser, or the Internet Explorer browser on the clients worked successfully File directories were constructed, and these directories could be shared with specific users or user groups [26], [32]. Image files (PNGs) and text files (.txt) were uploaded and downloaded into/from public directories, private directories, and existing workspaces. Files of both types could be locked or unlocked by users. The Content Service Search function worked as expected; specific content shared between emails,

image files, text files, and workspace documents could all be found. For example, a search of all directories for the word ‘FireFox’ revealed all four files located on the Oracle database: a PNG image file titled ‘FireFox.PNG,’ an email titled ‘FireFox,’ and a Microsoft Office Word 2003 document titled ‘FireFox.’ successful.

The Oracle Content Services web browser application operated without error when tested XTS-400 as a proxy. The test results in this configuration were identical to those observed when the clients were directly connected to the OCS server.

3. Oracle Calendar (web browser)

The Oracle Calendar application performed as expected on both OCS Clients directly connected to the OCS Server. The Oracle Calendar application could be accessed via a web browser, and could be used to set an appointment (meeting) on the Oracle User’s calendar (as described in Appendix B). Both the FireFox browser, or the Internet Explorer browser on the clients worked successfully. Appointments between individual users, appointments establishing web conferences, and the scheduling of tasks were all completed without error [26], [32]. Email notification of calendar appointments worked as expected.

When Oracle Calendar was tested on the OCS Clients using the XTS-400 as a proxy, the results were identical to those observed when the clients were directly connected to the OCS server. The application operated without error in this configuration.

4. Oracle Real-Time Collaboration: Web Conferencing (web browser)

The Oracle Real Time Collaboration Web Conferencing application performed as expected on both OCS Clients directly connected to the OCS Server. The Oracle Real-Time Collaboration Web Conferencing application could be accessed via a web browser, and could be used to create an instant web conference (as described in Appendix B).Both the FireFox browser, or the Internet Explorer browser on the clients worked successfully. Once a conference was started, users could chat, share documents, delegate and transfer speaker/presenter roles, transfer control of their own client workstation to other users, use

the Oracle-based electronic whiteboard, and communicate voice information from user to user via Voice Over Internet Protocol (VOIP) [26], [30], [32].

The Oracle Real Time Collaboration (RTC) Web Conferencing application did not function correctly when tested on the OCS Clients using the XTS-400 as a proxy. Although conferences could be scheduled in the Oracle Calendar, an error message stating 'System Error: Unable to connect to the server' would appear when the user attempted to launch a conference. Instant web conferences had the same error message. Figure 10 provides a screen shot image capture of this RTC error. The Diagnostic Test in the Oracle RTC application New User configuration console stated that, although the system was properly configured, the connectivity with the OCS server was inadequate to maintain a web conference [26], [30]. Figure 11 provides a screen shot image of the RTC Diagnostic Report recorded.

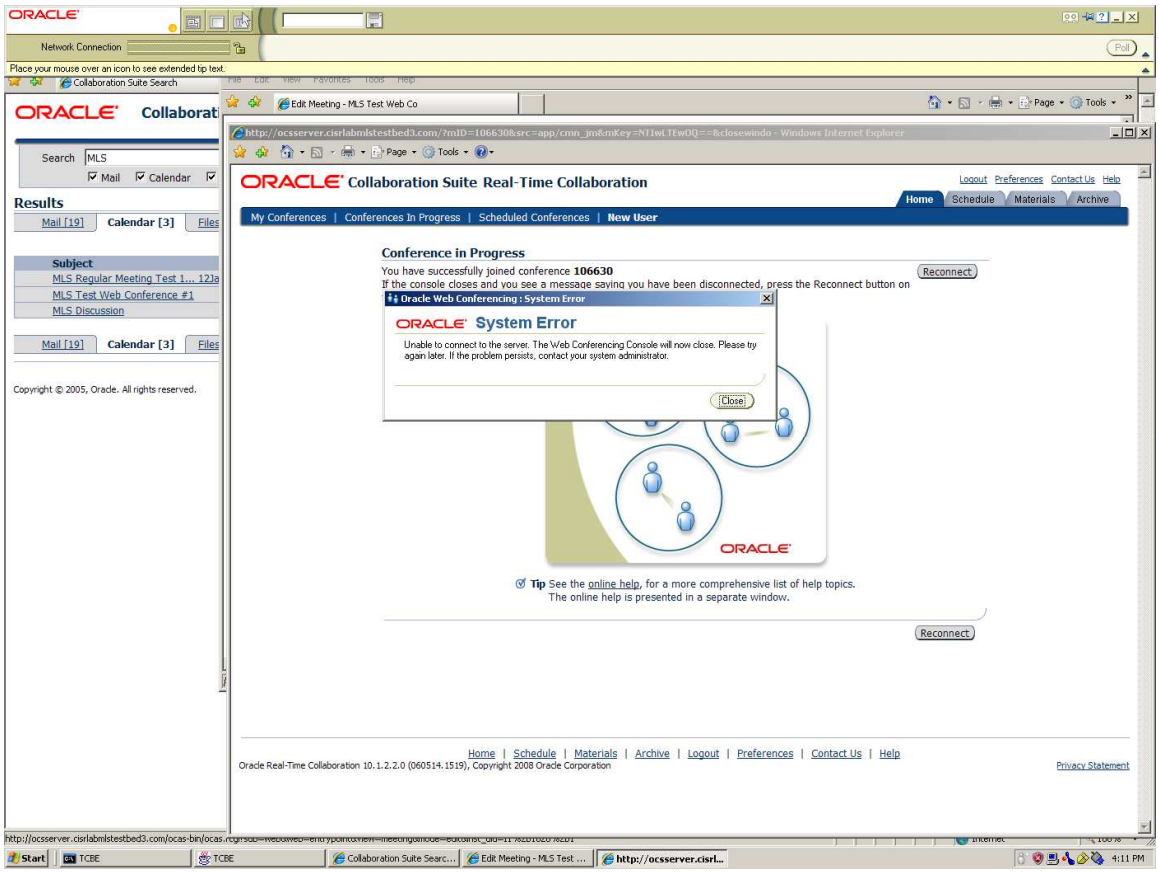

Figure 10. Oracle Collaboration Suite Real Time Collaboration application with System Error (Unable to Connect).

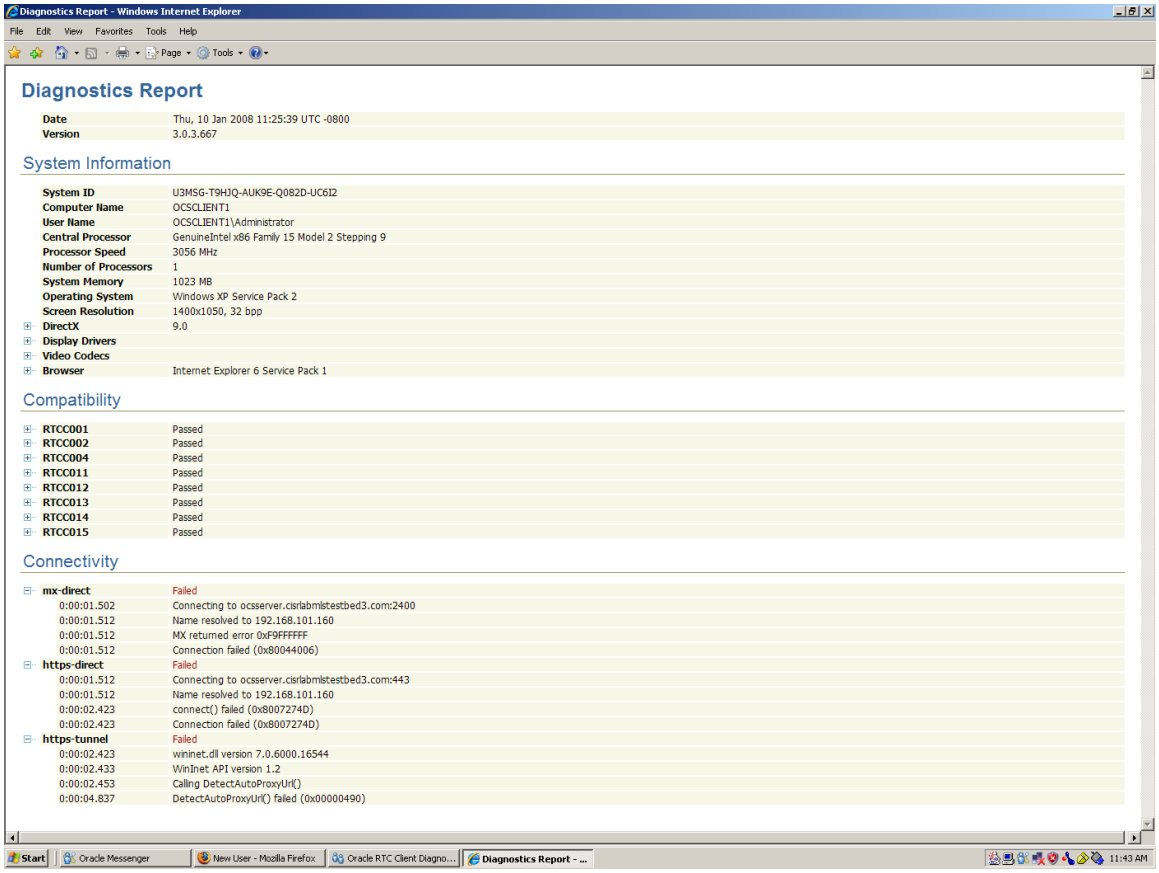

Figure 11. Oracle Collaboration Suite Real Time Collaboration (RTC) Diagnostic Report with Connectivity Failure.

5. Oracle Real Time Collaboration: Instant Messenger (rich media client)

The Oracle Real Time Collaboration Instant Messenger rich media client could not connect to the OCS server from either a client (OCS Client 1 or OCS Client 2), or the OCS Server itself. Since this application could not be made to function on the clients directly connected to the server, this application was not tested using the XTS-400 as a proxy. Section A of Chapter VII provides a further discussion regarding why this application was not tested in the second phase of testing.

6. Oracle Workspaces (web browser)

The Oracle Workspaces application performed as expected on both OCS Clients directly connected to the OCS Server. The Oracle Workspaces application could be

accessed via a web browser, and a workspace could be created for the Oracle User (as described in Appendix B). Both the FireFox browser, or the Internet Explorer browser on the clients worked successfully. File content in individual workspaces could be added or removed. Multiple users could be assigned to a workspace, and access roles assigned to different users could be changed or revoked [26]. Interactions between the Oracle Workspace application and these other (Oracle) applications in this configuration were transparent and seamless; the other Oracle applications functioned as designed when utilized by the Oracle Workspace application.

For the most part, Oracle Workspaces functioned as expected when tested on both clients at the single level while using the XTS-400 server as a proxy. Workspace administrators could add/remove file content, post discussions, assign users, modify user access roles, schedule conferences, and send various email notifications to workspace members. The only error was associated with starting web conferences inside workspaces. As noted in the part 4 of this Section, web conferences scheduled inside a particular workspace would fail at startup. The error message 'System Error: Unable to connect to the server' would appear when a user would attempt to join a scheduled web conference.

7. Oracle Discussions

The Oracle Discussions application performed as expected on both OCS Clients directly connected to the OCS Server. The Oracle Discussions application could be accessed via a web browser, and could be used to start a discussion thread in an existing Oracle User workspace (as described in Appendix B). Both the FireFox browser, or the Internet Explorer browser on the clients worked successfully. Users posted messages, shared files, replied to existing threads as expected [26].

The Oracle Discussions application operated without error when tested at a single level while using the XTS-400 as a proxy. The test results in this configuration were identical to those observed when the clients were directly connected to the OCS server.

8. Oracle Content Services: 'Oracle Drive' (rich media client)

The Content Services 'Oracle Drive' rich media application performed as expected on both OCS Clients directly connected to the OCS Server. The 'Oracle Drive' downloadable application was installed successfully on both clients, and could be used to upload a file from the client's desktop file directory into the Oracle User folder (as described in Appendix B). File directories could be opened, and image files (PNGs) and text files (.txt) were uploaded and downloaded into/from public directories, private directories, and existing workspaces [26], [32].

The 'Oracle Drive' also operated without error when tested at a single level while using the XTS-400 as a proxy. The test results in this configuration were identical to those observed when the clients were directly connected to the OCS server.

9. SMTP Mail Exchange (with Linux server)

Prior to single level testing, the OCS Server could not be configured to exchange email with another (directly connected) external server using Simple Mail Transport Protocol (SMTP) on Port 25 [39], [41]. The SMTP_Inbound and SMTP_Outbound relay settings were modified on the OCS Server to include the domain name (ocsserver1.cisrlabmlstestbed3.com) of another identical OCS Server, 'OCS Server1.' These settings were modified using the Mail Application page on the Oracle Application Control Console portal. After modification, a simple email test was conducted using the Oracle Web Mail application. An Oracle user on ocsserver1 (cprince@ocsserver1.cisrlabmlstestbed3.com) would attempt to send email to an Oracle user located on ocsserver (cmgilkey@ocsserver.cisrlabmlstestbed3.com). Unfortunately, the sending server (ocsserver1) could not connect with the receiving server (ocsserver). An error message would appear, stating 'Error Sending Message. Could not connect to SMTP host, Connection refused.' The error message appeared again when the servers changed roles (ocsserver attempted to send an email to ocsserver1). Figure 12 provides a screen shot image of the Oracle Web Mail error message observed. The Oracle Mail Administrator's Guide indicated that the 'relay' field was not configured correctly on the receiving server's SMTP_Inbound Mail Application Control Console page [33]. The

SMTP settings were rechecked, and appeared to be correct according to the Oracle Mail Administrator's Guide. Unfortunately, subsequent retests yielded the same error message. Since the OCS Server could not be properly configured to exchange email with an external server using SMTP, this test was not conducted at a single level using the XTS-400 as a proxy.

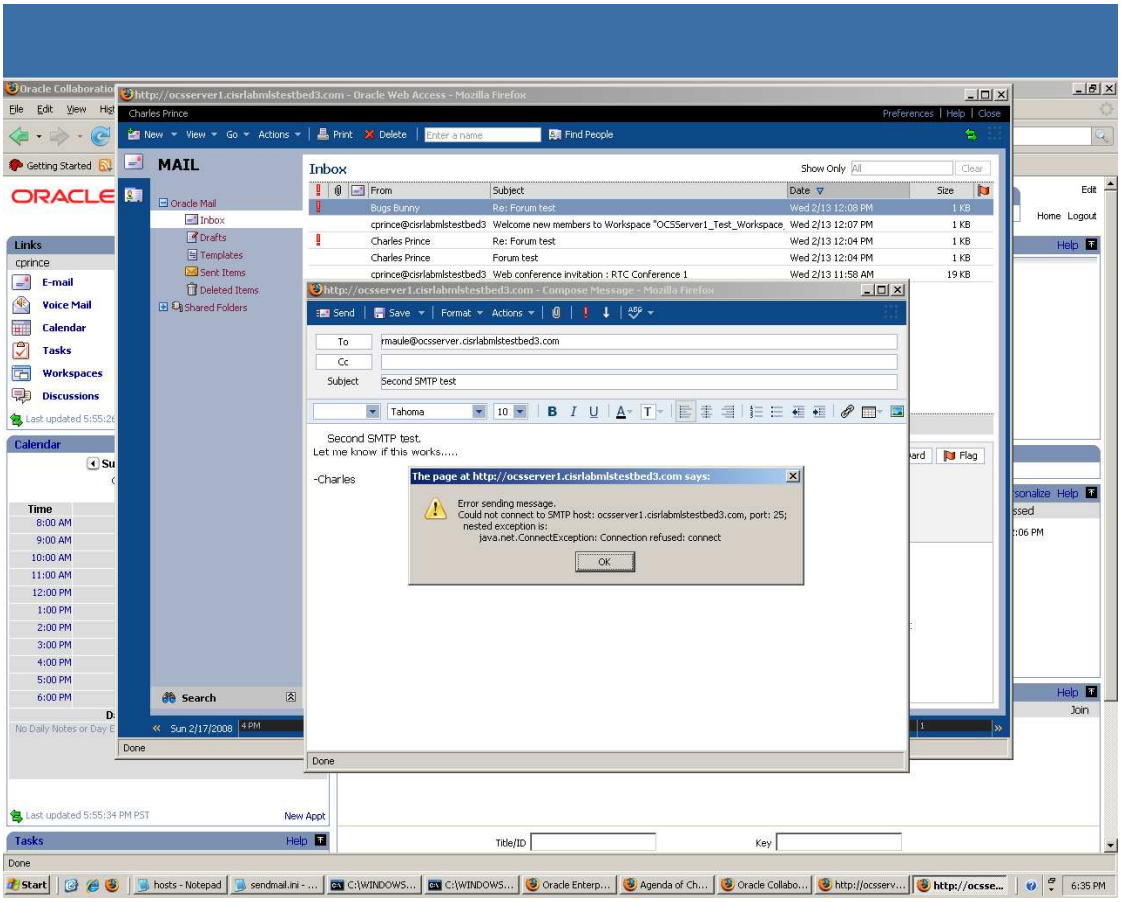

Figure 12. Oracle Collaboration Suite Simple Mail Transport Protocol (SMTP) Connection Refused.

E. SUMMARY

This chapter presented the components tested, the functional test plan, and the test results in support of this project's proof of concept: the integration of a SOA into a multilevel secure environment. All tests were completed, and a high-level description of the results was recorded in Section B of this chapter. These results are evaluated and interpreted in the next chapter.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. ANALYSIS AND DISCUSSION

This chapter has two purposes: to provide an analysis of the experiment's test results, as described in the previous chapter, and also to discuss some of the design and/or configuration issues that were noted during installation and testing. The first section of this chapter is devoted to analysis of the test results. The second section describes some of the issues noted in this experiment.

A. ANALYSIS OF TEST RESULTS

This section provides an analysis of the test results, as detailed in the previous chapter. The discussion in this section is divided into two parts, one for first phase of testing ('direct connection' testing), and a second part for the subsequent testing phase (single level testing, using the XTS-400 as a proxy).

1. Direct Connection Testing Analysis

This phase of testing was expected to have few, if any errors, since it was conducted on an OCS server directly connected to two clients, and was isolated from other external applications, application servers, and/or network traffic. As described in the previous chapter, the only OCS application component to fail in this phase of the testing was the Real Time Collaboration Instant Messenger client (or 'RTC Messenger').

At this point, it was also noted that the OCS Server could not be configured to exchange email with another (directly connected) external mail server using Simple Mail Transport Protocol (SMTP) on Port 25. A successful email exchange of this kind would be beneficial to the MYSEA testbed. The simulated SIPRNet enclave of the MYSEA environment included a Linux operating system-based SMTP server that had been successfully implemented and tested on the simulated multilevel testbed for access via the XTS-400 [8]. Given that (a) this test was outside the scope of the original function

test plan in Chapter III, and (b) the SMTP settings could not be configured while directly connected, this test was not conducted at the single level using the XTS-400 as a proxy. It has been reserved for future work.

Although it would have been an added benefit to the experiment to have this third party client operational, this failure did not affect the proof of concept goal of this project since it (a) utilized a TCP/IP protocol outside the configuration settings of the experiment, (b) contained a protocol that was not mandatory per DoD SOA standards, and (c) had little bearing on future 'next generation' SOA designs [4], [36], [37], [40].

As discussed in Section B of Chapter IV, the RTC Messenger was one of two application components deployed for testing that was not HTTP-based (the WebDav-based 'Oracle Drive' rich media client was the other exception). The RTC Messenger client was built on the Jabber Instant Messenger [30]. Jabber is an open source instant messaging application based on the Extensible Messaging and Presence Protocol (XMPP) [42], [43]. As described in RFC 3920, "XMPP is a protocol for streaming Extensible Markup Language (XML) elements in order to exchange structured information in close to real time between any two network endpoints" [42]. XMPP clients use Transport Control Protocol (TCP) to connect directly to a server, typically over Port 5222 [42]. Even if this client was functional at this stage, it would probably not function in later phases of testing, since the OCS server was only configured to handle Port 80 (HTTP) requests from the OCS Clients. Additionally, XMPP was not listed as a 'mandatory' protocol in compliance with the SOA standards set forth by the DoD I.T. Standards Registry [4]. In plain language, XMPP is not a required to be a functional protocol for a SOA operating in the DoD. Between the limitations of the experiment and the actual DoD SOA compliance standards, additional troubleshooting and testing with this application became unnecessary.

Randy Maule noted that the RTC Messenger was a legacy tool that was included as a plug-in client for the Oracle Collaboration Suite SOA [40]. In this context, the term 'legacy' refers to the fact that the RTC Messenger was an existing rich media client added to the software suite of SOA applications. RTC messenger was included to provide additional functionality for the core XML-based applications built specifically as 'SOA'

components for the Oracle Collaboration Suite 10g (e.g., mail, discussions, content services, etc.). He added that most of the legacy-based OCS tools (like the RTC Messenger) were categorized as end of lifecycle products by Oracle [34], [40]. Support for such tools would eventually disappear, as the 'next generation' Oracle SOA (which will replace the Oracle Collaboration Suite 10g) would not include legacy based RTC tools [38], [40]. Tools like the RTC Messenger would either be redesigned as core XML-based 'SOA-components,' or not included in the 'next generation' Oracle SOA Fusion Middleware release [36], [37], [40]. For these reasons, future experiments should use the more comprehensive, 'next-generation' SOA as it becomes available (vice the OCS 10g, which is at the end of its product life cycle) [40].

2. Analysis of Testing the OCS Services at the Single Level using an XTS-400 as a Proxy

As detailed in the previous chapter, there were two errors and anomalies noted when the OCS Clients were tested in a single level configuration while using the XTS-400 acting as a proxy: (a) failure of the RTC applications (Web Conference and Instant Conference to startup, and (b) the inclusion of HTML source code in the body of a newly-generated Oracle Mail email.

The reason why the RTC web conferencing application components would not function was discovered in the Oracle RTC Administrator's Guide [24]. The RTC components would only function over a TCP/IP connection via Hyper-Text Transfer Protocol (HTTP) with Secure Socket Layer Encryption (SSL) over Port 443. Since the OCS Server was not configured to handle requests over HTTPS (Port 443), an RTC connection could not be established between the OCS Clients and the OCS Server [24]. Figure 13 describes this limitation: in the diagram, User 3 can access the Oracle Real-Time middle tier applications only via Port 443 when using a proxy server over a TCP/IP connection [24].

A remedy for this problem was not sought. As discussed in Chapter IV, configuration of an OCS server for HTTPS is non-trivial and must be initiated at the beginning of the OCS software installation [40]. Also, this reconfiguration would be

outside the original scope of the test plan. For future work, OCS Servers utilized in the MYSEA testbed should be initially configured for HTTPS (with 128-bit encryption) vice HTTP, to both (a) meet the Defense Information Systems Agency (DISA) security requirements for DoD web clients, and (b) to allow the Oracle RTC applications to function [4], [30].

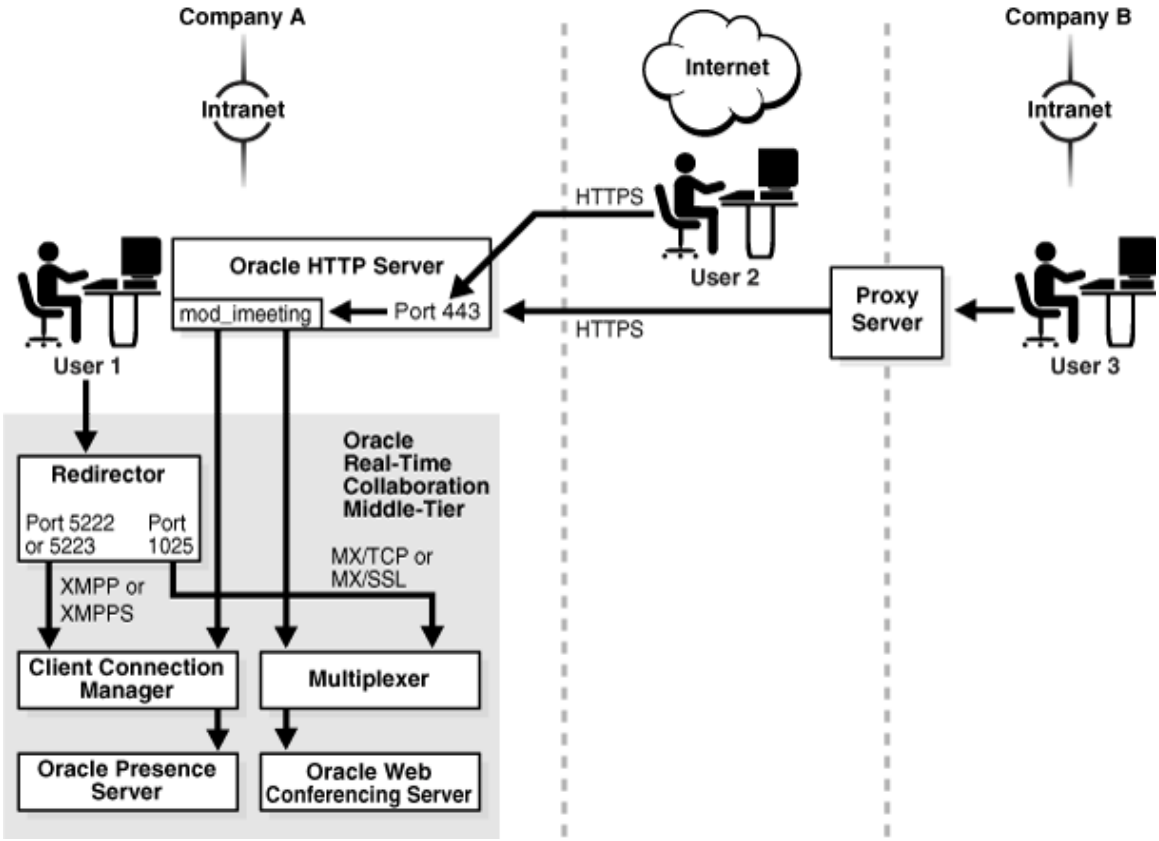

Figure 13. Remote access of Oracle Real-Time Collaboration Middle-tier applications via HTTPS [from [24]].

The findings of this experiment indicate that the Oracle applications tested should function in the actual MYSEA testbed. Future iterations of this testing, should be conducted over TCP/IP connections utilizing HTTPS vice HTTP, to meet DISA's security requirements. Using HTTPS will also permit the use of the Oracle RTC components. There is no reason why all of the Oracle applications tested in this experiment (including the RTC application components) should not function in a similar

configuration using HTTPS, particularly since the XTS-400 servers have been successfully configured and tested to act as a proxy over TCP/IP connections at Port 443 (HTTPS).

B. CONFIGURATION ISSUES

This section discusses configuration issues experienced in this experiment. These issues were specific to either (a) the installation of the OCS software, or (b) to the system performance of the OCS server itself. The issues presented were experienced using the functional OCS Single Server configuration specified in Chapter V.

Installation of the OCS software suite expended far more time than initially anticipated. Due to miscommunication regarding the specificity of the OCS server's domain name (http://ocsserver.cisrlabmlstestbed3.com), the first six installation attempts failed. Other configuration issues increased the total time associated with the installation to 42 hours expended. Appendix A (Installation Procedures) was written specifically to reduce future frustration and time associated with the installation process. Charles Prince, an engineer on the CISR staff, conducted an independent verification of Appendix A. He expended a total of 22 hours installing the software due to issues associated with (a) errors in the first draft of the Appendix, and (b) the length of time required to install the software. Future iterations of this experiment should note the nontrivial, significant time required to install the software.

The system performance of the OCS server, while not critical to the success of the experiment, could be described as 'sluggish,' at best. The hardware chosen as part of the functional test plan, as described in Chapter V (a Dell Dimension 4600 with a Pentium IV 3.0 Ghz processor, 2 GB of RAM, 20 GB of available hard drive space) met the minimum requirements for a Windows Server 2003 Service Pack 2 Single Server configuration, listed in the Oracle Collaboration Suite 10g Installation Guide: 'at least 2 GB of RAM, 17.6 GB of available hard drive space, and a Pentium 2 Ghz processor or greater [27].' During the installation of the software, the OCS installation software reported: 'system hardware requirements met [27].' Following installation of the software, the system performance of the OCS server degraded significantly, particularly

in processes associated with the OCS software itself. Depending on how long the components of the OCS had been running, the Oracle Collaboration Suite Portal page would frequently take one to two minutes to load on an OCS Client. Worse, some applications would fail to load if the OCS components had been running for more than eight hours. For example, after leaving the OCS server components up and running for more than 48 hours, the Oracle Mail application would not load. After performing a manual shutdown and restart of both the OCS Infrastructure and the OCS Application Components, the Oracle Mail application functioned correctly. These problems were attributed to insufficient RAM [40].

Because of the apparent lack of RAM, another performance issue was noted in relation to Windows Page File sizes. After installing the OCS software, the Windows operating system displayed a warning that the default virtual page file employed by the system (3069 MB) should be increased to 4096 MB. After the page file size was increased (as detailed in Figure 14), the warnings ceased to arise; system performance, however, remained slow and did not improve measurably. After observing these issues, Randy Maule strongly suggested that future iterations of this experiment utilize an enterprise-class server with at least 4 GB of RAM [33].

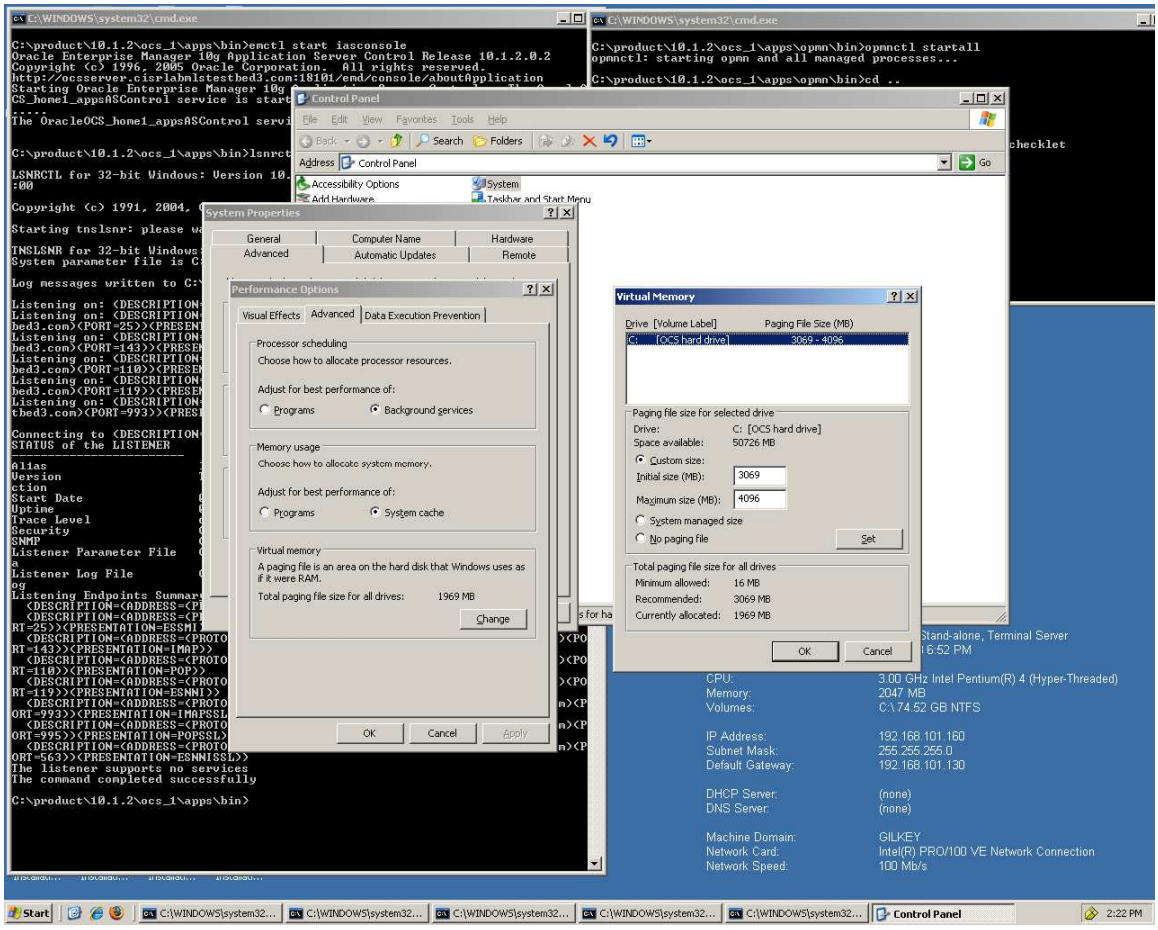

Figure 14. Windows Page File (virtual memory) modification on the OCS server.

C. SUMMARY

This chapter presented an analysis of the test results from Chapter V, along with a discussion of both installation and performance issues associated with the OCS server used in this experiment. The test results analysis are encouraging and support future development of SOA software in the MYSEA testbed. The points made in this chapter reinforce the need for the future work described in the next chapter.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. CONCLUSIONS AND FUTURE WORK

This chapter will state the project conclusions, and suggest possible work for the future.

A. CONCLUSION

As described in Chapter III, the goal of this thesis is to determine the feasibility, or proof of concept, of incorporating a web-based SOA software suite into a multilevel environment. This goal has been successfully achieved through the integration of a SOA software suite and its web-based applications onto a copied segment of the MYSEA testbed. The tests described in Chapter V confirmed that the HTTP web browser applications on an OCS 10g server are fully functional at a single level using an MLS server (the XTS-400) as a proxy. Only one web browser application did not to function at a single level (Oracle Real-Time Collaboration), but this was because the OCS 10g server was not configured for HTTPS (the Oracle Real-Time middle tier applications can only be accessed via Port 443 when using a proxy server over a TCP/IP connection) [30]. The findings of this experiment indicate that SOA web browser applications should function in the actual MYSEA testbed.

B. FUTURE WORK

This section includes six major areas identified during this experiment that warrant future work. These areas are: HTTPS support, connection with an external SMTP mail server, enterprise-class server deployment, multi-computer deployment, Oracle (Next Generation) Fusion Middleware, and ‘MLS Aware’ SOA applications.

1. HTTPS Support

As discussed in Chapter VI, OCS 10g servers utilized in the MYSEA testbed should be initially configured for HTTPS (with 128-bit encryption) vice HTTP, to meet the Defense Information Systems Agency (DISA) security requirements for DoD web clients, and to allow the Oracle Real-Time Collaboration (RTC) web browser application

to function [4], [30]. Since the XTS-400 servers have been successfully configured and tested to act as a proxy over TCP/IP connections at Port 443 (HTTPS), there is no reason why all of the OCS 10g web browser applications (including the RTC web browser application) should not function in a similar configuration using HTTPS. Future work should include using HTTPS (instead of HTTP).

2. Connection with an External SMTP Mail Server

As discussed in Chapter V, the OCS 10g server could not be configured to exchange email with another (directly connected) external mail server using Simple Mail Transport Protocol (SMTP) on Port 25 [13]. Enabling an email exchange between an external (SMTP) mail server and the OCS 10g server would be very beneficial considering that there is a Linux operating system-based SMTP server that had been successfully implemented and tested on the simulated multilevel testbed [8]. Future work in this area might be simple, considering that (a) only the SMTP settings on the OCS 10g need to be modified, and (b) the OCS 10g server provides a browser-based control panel (the Oracle Collaboration Suite Application Console Control) to change or reconfigure `SMTP_Inbound` both `SMTP_Outbound` [39]. Future work should include testing the OCS 10g server with an external SMTP mail server.

3. Deployment of the OCS Software Suite: Enterprise-Class Server

As noted in Chapter VI, the performance of the OCS 10g server, although adequate, was less than desirable. Future single computer deployments should utilize an enterprise-class server with at least 4 GB of RAM [40].

4. Deployment of the OCS Software Suite: Multi-computer Deployment

As described in Chapter II, there are several multi-computer deployment options with the OCS 10g software suite. These multi-computer configurations deploy the key Oracle components (the Oracle Applications Tier, the Oracle Internet Directory, the Oracle Database, and the Oracle Infrastructure Tier) between two or more computers. [16]. The Oracle Deployment Guide recommends these multi-computer configurations for user groups greater than 1,000 [28]. Future iterations of this experiment might consider deploying the OCS 10g (or other) software suite in a multi-computer

configuration, especially since the TACFIRE research portal has been configured in a multi-computer configuration to meet the demand of large numbers of users.

5. Oracle Fusion Middleware

As described in Chapter II, the next evolution of the Oracle Application Server (Oracle Fusion Middleware) will unify all of the Oracle applications under a single set of 'web-service' based standards [36], [37], [40]. Additionally, this 'application unification' will now provide users with web service tools identical to those provided in Web 2.0 technology [38]. Future iterations of this experiment should implement Oracle Fusion Middleware to test these web service tools, and to take advantage of the fact that there will be no residual legacy based applications (like in OCS 10g) [40].

6. 'MLS Aware' SOA Applications

As discussed in Chapter II, several protocols have been hosted on MLS high assurance servers as 'MLS Aware' applications. Extensive modifications allow these applications to reside and function as single level applications on the MLS server itself. By being 'MLS Aware,' the application is able to read down to appropriate information at lower security levels. SMTP, IMAP, and WebDAV are some of the protocols that have been adapted to be as 'MLS Aware' in the MYSEA testbed [7], [8], [31]. Incidentally, these protocols are all actively used by certain individual OCS 10g applications: both IMAP and SMTP are used by the Oracle Web Mail application, and the Oracle File Content Services uses WebDAV [33], [39]. Additionally, the OCS 10g software suite provides a browser-based control panel (the Oracle Collaboration Suite Application Console Control) to modify and reconfigure the SMTP and IMAP settings of the OCS 10g server. Creating an email exchange between the 'MLS Aware' SMTP mail server and the OCS 10g mail server might be as simple as modifying the SMTP settings on the Oracle Collaboration Suite Application Control Console. Future iterations of this experiment might focus on one of two goals: (a) to analyze the requirements for creating either an SMTP based or an IMAP based email exchange between the 'MLS Aware' mail server and the OCS 10g mail application, or (b) to determine what modifications should be made to either the Oracle Mail application or the Oracle File Content application to make them 'MLS Aware' applications actually capable of residing on the MLS server.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A: INSTALLATION PROCEDURES

This appendix describes procedures for installing the Oracle Collaboration Suite (OCS) 10g, version 10.1.2, in the Monterey Security Architecture (MYSEA) simulated multilevel testbed.

Since testing the performance of the OCS applications is not part of the analysis, the OCS is installed as a Single-Computer Installation. In this Single Computer configuration, the Oracle Database, the Oracle Applications tier, and the Oracle Infrastructure tier all reside on the same computer. This appendix follows the installation procedures outlined in the Oracle Collaboration Suite Installation Guide 10g Release 1 (10.1.2). File names and directory locations for specific Oracle installation files (Oracle Database files, Oracle Infrastructure files, and the Oracle Application files) are annotated in this section. The Oracle Namespace in the Oracle Internet Directory is specified as `cisrlabmlstestbed3.com` for correct operation in the MYSEA simulated multilevel testbed.

Testing the OCS will occur in two phases. In the first phase of testing, the OCS server is initially configured to operate in an intranet isolated from the simulated MYSEA multilevel testbed. In this setup, the OCS server is directly connected to the OCS clients via a single switch. This configuration is established to check the correct behavior of the OCS applications, and to establish an 'expected result' for each OCS application prior to simulated multilevel testing. In the second phase of testing, the OCS clients will access the OCS server (and test the OCS applications) via a multilevel secure XTS-400 server acting as a proxy. This testing will establish the functionality of web-based OCS applications at the single level (on a simulated MLS environment). Additionally, the XTS-400 server will be connected to a Linux Mail server residing on the simulated SIPR enclave. The OCS Server and the Linux Mail server will be tested to see if they can exchange email via Simple Mail Transfer Protocol (SMTP).

To eliminate moving hardware components multiple times, all of the equipment used in both phases of testing will be setup first, as outlined in Section A: Initial

Hardware Setup. Once the procedures in Section A are completed, the OCS server will be connected directly to the two OCS Clients via a single switch, as outlined in Section B: Connecting the OCS Server Directly to the OCS Clients. Upon completion of testing the OCS applications in this configuration (direct connection testing), the OCS server will be connected to the switch to which the XTS-400 is connected, and the OCS clients will connect to the OCS server via the XTS-400 as a proxy (see Section K: Connecting the OCS Server to the Simulated MLS Environment).

Each section of this Appendix is dependent on previous sections, and the installation procedures should be followed in order from start to finish. The Internet Protocol (IP) addresses of the OCS servers and OCS clients will be changed to suit various test configurations. However, the IP addresses of the XTS-400 server and the Linux Mail Server residing on the simulated SIPR enclave will not change. Instructions for testing the individual applications (services) of the installed software are provided in Appendix B: Test Procedures.

The instructions herein also reference the Secure Attention Key (SAK). The SAK is invoked by a user at the virtual Trusted Path Extension device on the client by simultaneously pressing the 'Ctl,' the 'Alt,' and the 'Print Screen' keys. This process permits special trusted commands specific to the MLS server, including the *sl* command (which is used to set the security level of a particular session).

A. INITIAL HARDWARE SETUP

The installation steps listed in this section outline the initial hardware setup to support the test procedures described in Appendix B. The network topology consists of five computers connected via three switches as shown in Figure A, Network Topology for Initial Installation, below.

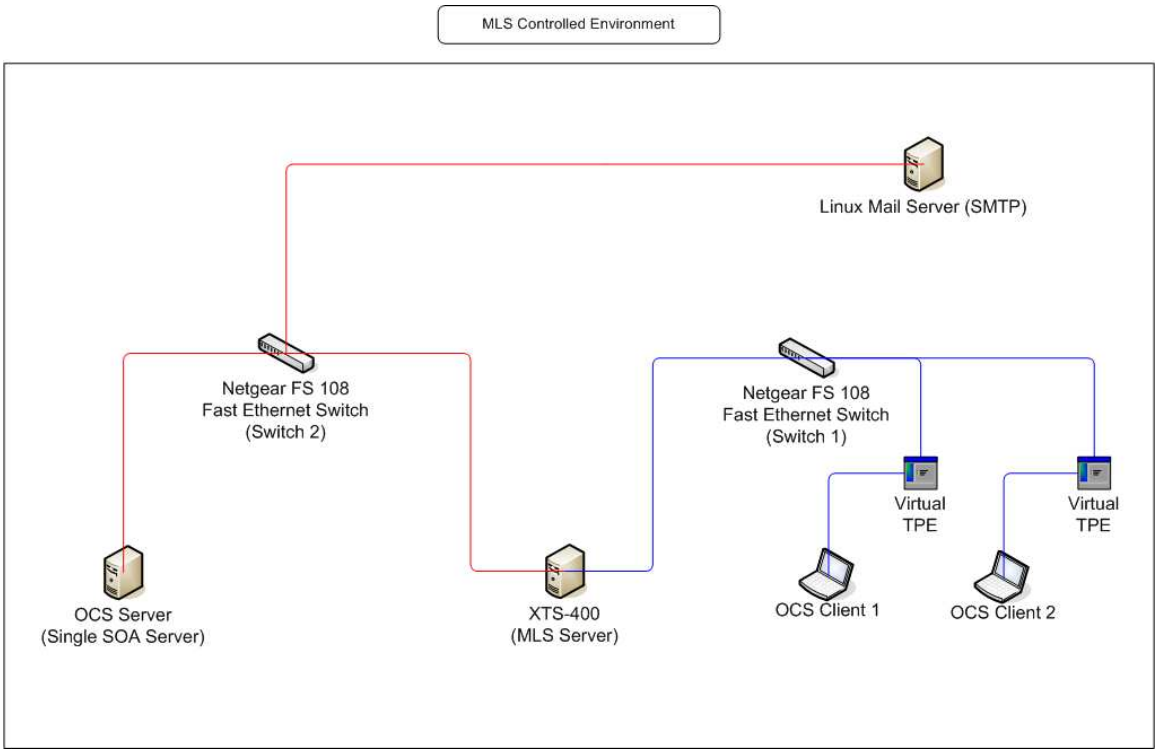

Figure A. Network Topology for Initial Installation.

The first two computers are labeled OCS Clients, and run Windows XP (Service Pack 2 installed). Each of these computers will have (at the minimum) a Pentium III 1.0 Ghz processor, 1 GB of RAM, 10 GB of available hard drive space, input jacks for a microphone, input jacks for headphones, and an Ethernet connection jack. Both OCS Clients will have Internet Explorer 7.0 with the Java Runtime Environment Version 6, Update 3 plugin installed. Mozilla FireFox 2.0.0.11 is also installed on both clients. All installation procedures will use the Windows *Administrator* account for both OCS Clients. OCS Client 1 and OCS Client 2 will both have a 'Virtual' Trusted Path Extension file (`tcbe.exe`), known as the 'Trusted Computing Base Extension' program, installed on the Windows desktop.

The third computer is the OCS Server. The OCS Server runs Windows Server 2003 (Service Pack 2 installed). This computer will have (at the minimum) a Pentium IV 3.0 Ghz processor, 2 GB of RAM, 20 GB of available hard drive space, input jacks for a microphone, input jacks for headphones, and a Network Interface Card to support an Ethernet connection. The OCS Server will have Internet Explorer 7.0 with the Java

Runtime Environment Version 6, Update 3 plugin installed. Mozilla FireFox 2.0.0.11 is also installed on this computer. All installation procedures will use the Windows *Administrator* account for the OCS Server. This computer is setup in Windows Server 2003 as a standalone terminal server, with no server roles configured.

The fourth computer (the MLS server) is an XTS-400 running the STOP 6.1 operating system, including a minimum of two Network Interface cards. This computer will be configured by the CISR staff.

The fifth computer is the Linux (SMTP) Mail server residing on the simulated SIPR enclave of the simulated MLS environment. This server runs Red Hat 9, and will be configured by the CISR staff.

OCS Clients 1 and 2 are connected to the XTS-400 via the first switch (Switch 1 in Figure 1), a Netgear FS 108 Fast Ethernet switch. The XTS 400 is also connected to a to a Linux Mail server, which serves as an external mail application server. This Mail server serves as a simulation of the actual Linux Mail server residing on the simulated SIPRNet of the MYSEA testbed.

Note: Third party virus scanners, spam blockers, and firewalls, if present on OCS Server, OCS Client 1 or OCS Client 2, should be disabled. This is a requirement listed in Section 2 of the Oracle Collaboration Suite Administrator's Guide 10g Release 1 (10.1.2) [17].

B. CONNECTING THE OCS SERVER DIRECTLY TO THE OCS CLIENTS

The installation procedures listed in this section outline the steps to connect the OCS Server to the switch where the OCS Clients reside. Prior to executing these steps, the OCS Server should be powered down. See Figure 10, Network Topology for Initial Installation, in Section A of this Appendix, for a description of what the network topology should look like prior to executing these steps.

Step 1. Disconnect the Ethernet cable from the OCS Server to Switch 2.

Step 2. Reconnect the Ethernet cable from the OCS Server to Switch 1 (the switch where both OCS Client 1 and OCS Client 2 reside).

Step 3. Verify that the changes made in Steps 1 thru 2 resemble the topology illustrated in Diagram B, Network Topology for Direct Connection Testing, below. **Note:** The Netgear FS 108 Fast Ethernet Switch in Figure B is the same switch as Switch 1 in Diagram A.

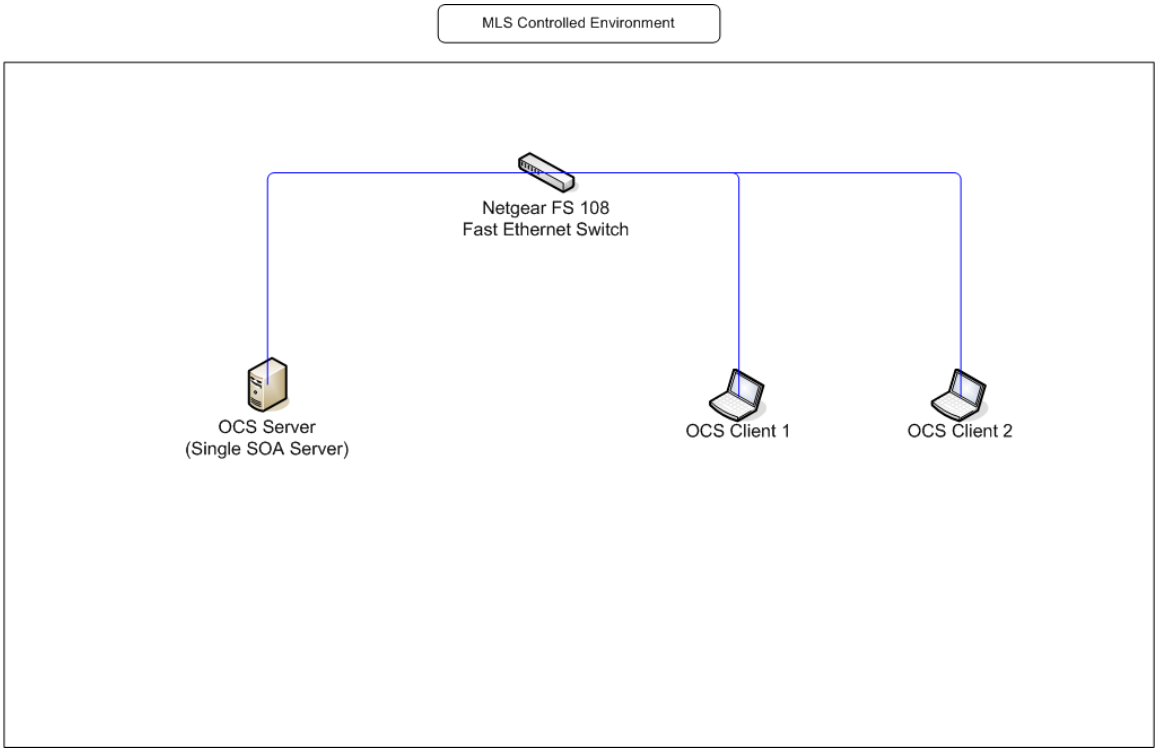

Figure B. Network Topology for Direct Connection Testing (Network 192.168.101.X).

C. SETTINGS FOR THE XTS-400 MLS SERVER AND THE LINUX MAIL SERVER

The IP Address for the XTS-400 MLS Server is `192.168.101.130.` The CISR lab staff will configure the IP Address of the XTS-400 server. During the MLS testing, the XTS-400 will serve as the Proxy server (`192.168.0.130`) for the OCS Clients. Since only single level testing (across the simulated SIPRNet) is conducted, these settings do not change and remain static throughout testing.

The IP Address for the Linux Mail server on the simulated SIPR enclave is `192.168.101.2.` The CISR lab staff will configure the IP Address of the Linux Mail

server. Since only single level testing (across the simulated SIPRNet) is conducted, these settings do not change and remain static throughout testing.

D. SETTINGS FOR WINDOWS AND WEB BROWSER APPLICATIONS FOR DIRECT CONNECTION TESTING

Prior to installing the OCS software on the OCS Server, several changes will be completed on the OCS Server and the OCS Clients. These changes include rewriting the `hosts` file, modifying IP addresses, turning the Windows Firewall off, and altering the settings on both the Mozilla FireFox web browser and the Internet Explorer 7.0 web browser. The installation steps listed in this section support the configuration requirements for the Direct Connection Testing steps outlined in Appendix B (for the installation steps to configure the computers for MLS testing, see Section G of this Appendix). All of the procedures listed in this section are to be completed under the *Administrator* account in Windows.

Step 1. Rewrite the IP addresses and domain names listed in the hosts file on the OCS Server. Login as the Windows Administrator on the OCS Server. Open the hosts file located in the C:\\WINDOWS\system32\drivers\etc\ directory. Replace the existing text with the following**:**

```
127.0.0.1         localhost.localdomain               localhost
192.168.101.164   ocsserver1.cisrlabmlstestbed3.com   ocsserver1
192.168.101.160   ocsserver.cisrlabmlstestbed3.com    ocsserver
```

Note: Once the `hosts` file on the OCS Server has been configured, it will not change through either Direct Connection Testing and/or MLS Testing.

Step 2. Rewrite the IP addresses and domain names listed in the `hosts` file on OCS Client 1 and on OCS Client 2. Login as the Windows *Administrator* on OCS Client 1. Open the `hosts` file located in the `C:\\WINDOWS\system32\drivers\etc\` directory. Replace the existing text with the following:

```
127.0.0.1         localhost.localdomain               localhost
192.168.101.164   ocsserver1.cisrlabmlstestbed3.com   ocsserver1
```

```
192.168.101.160   ocsserver.cisrlabmlstestbed3.com    ocsserver
192.168.101.161   ocsclient1.cisrlabmlstestbed3.com   ocsclient1
192.168.101.162   ocsclient2.cisrlabmlstestbed3.com   ocsclient2
```

Repeat these actions on OCS Client 2

Step 3. On the OCS Server, open the Network Connections tab in the Windows Control Panel. Right click the Local Area Connections icon, and select 'Properties.' In the window that appears, click on the 'Internet Protocol (TCP/IP)' icon, and click on the 'Properties' tab. Select 'Use the Following IP Address,' and depending on which computer is being modified, set the IP address as follows:

- OCS Server (ORACLE) `192.168.101.164`
- OCS Client 1 `192.168.101.161`
- OCS Client 2 `192.168.101.162`

Note: Table 2 provides a detailed description of the IP Addresses and the Default Gateway settings on the OCS Clients and the OCS Server for both (a) when the clients are directly connected to the OCS Server (Direct Connection testing) and (b) when the OCS applications are being tested at the single level using the XTS- 400 as a proxy server.

Table 2 IP Addresses and the Default Gateway settings on the OCS Clients and the OCS Server

	Direct Connection Testing IP Address	Single Level using XTS-400 as a Proxy IP Address	Direct Connection Testing Default Gateway	Single Level using XTS-400 as a Proxy Default Gateway
OCS Server1	192.168.101.164	192.168.101.164	192.168.101.130	192.168.101.130
OCS Client1	192.168.101.161	192.168.0.31	192.168.101.130	192.168.0.130
OCS Client2	192.168.101.162	192.168.0.32	192.168.101.130	192.168.0.130

In the ‘Subnet Mask’ field, enter `255.255.255.0.` In the ‘Default Gateway’ field, enter `192.168.0.130` (which is the XTS-400’s Proxy server address). Repeat these actions on OCS Client 1, and OCS Client 2.

Step 4. Ask the CISR Staff to verify that the `hosts` file on the XTS-400 includes the following entry:

```
192.168.101.164   ocsserver1.cisrlabmlstestbed3.com   ocsserver1
```

Step 5. Open the Control Panel on the OCS Server. Double click the ‘System Properties’ icon, and then click ‘Computer Name.’ Change the ‘Name’ field to `OCSSERVER1`. Click the ‘Change…’ button on the window. In the new window that appears, make sure the ‘Workgroup’ radio button under ‘Member Of:’ is selected, and that the ‘Workgroup’ field includes the title `WORKGROUP`. Above the ‘Member Of:’ area, click ‘More,’ and enter `cisrlabmlstestbed3.com` in the Domain field. Restart the computer when prompted.

Step 6. Open the Control Panel on the OCS Client 1. Double click the ‘System Properties’ icon, and then click ‘Computer Name.’ Change the ‘Name’ field to `OCSCLIENT1`. Click the ‘Change…’ button on the window. In the new window that appears, make sure the ‘Workgroup’ radio button under ‘Member Of:’ is selected, and

that the ‘Workgroup’ field includes the title `WORKGROUP`. Above the ‘Member Of:’ area, click ‘More,’ and enter `cisrlabmlstestbed3.com` in the Domain field. Restart the computer when prompted.

Step 7. Open the Control Panel on the OCS Client 2. Double click the ‘System Properties’ icon, and then click ‘Computer Name.’ Change the ‘Name’ field to `OCSCLIENT2`. Click the ‘Change…’ button on the window. In the new window that appears, make sure the ‘Workgroup’ radio button under ‘Member Of:’ is selected, and that the ‘Workgroup’ field includes the title `WORKGROUP`. Above the ‘Member Of:’ area, click ‘More,’ and enter `cisrlabmlstestbed3.com` in the Domain field. Restart the computer when prompted.

Step 8. On the OCS Server, open the Network Connections tab in the Windows Control Panel. Right click the Local Area Connections icon, and select ‘Properties.’ In the window that appears, click on the ‘Advanced’ tab. In the ‘Windows Firewall’ area, click the ‘Settings’ tab and disable the Windows Firewall/ICS Service. This is a requirement listed in Section 2 of the Oracle Collaboration Suite Administrator’s Guide 10g Release 1 (10.1.2) [17].

Step 9. On OCS Client 1, open the Network Connections tab in the Windows Control Panel. Right click the Local Area Connections icon, and select ‘Properties.’ In the window that appears, click on the ‘Advanced’ tab. In the ‘Windows Firewall’ area, click the ‘Settings’ tab. In the window that appears, select ‘Off (not recommended)’ and click ‘OK.’ Repeat these steps for OCS Client 2.

Step 10. Install FireFox 2.0.0.11 on both OCS Client 1 and OCS Client 2. Double click the file labeled ‘`Firefox Setup 2.0.0.11.exe`’ to begin the installation.

Step 11. On the OCS Server, double click on the Mozilla FireFox icon on the desktop. On the FireFox menu bar, click ‘Tools,’ and select ‘Options.’ Under the ‘Advanced’ section of the ‘Options’ bar, open the ‘Network’ tab. Click ‘Settings,’ and select ‘Direct connection to the Internet.’ Click ‘OK,’ the ‘Content’ tab on the ‘Options’ bar, and ensure that both ‘Enable Java’ and ‘Enable Javascript’ are checked.

Step 12. Repeat Step 8 on OCS Client 1 and OCS Client 2.

Step 13. On the OCS Server, double click on the Internet Explorer 7.0 icon on the desktop. On the Explorer menu bar, click ‘Tools,’ and select ‘Internet Options.’ On the ‘Intenet Options’ tab, click ‘Privacy.’ In the ‘Privacy’ area, uncheck ‘Turn Pop-up Blocker On,’ and set the ‘Settings’ bar to ‘Accept All Cookies.’

Step 14. On the ‘Intenet Options’ tab, click ‘Advanced,’ and ensure all of the following boxes remain **unchecked**:

- Enable Integrated Windows Authentication
- Enable native XML HTTP support
- HTTP 1.1
- HTTP 1.1 thru Proxy Connections
- SSL 2.0
- Check for signatures on Downloaded Programs

Step 15. Click the ‘Internet Options’ tab (as described previously in Step 13). Under ‘Connections,’ click ‘LAN Settings.’ Check the box marked ‘Automatically Detect Settings.’ Uncheck the box labeled ‘use a proxy server for your LAN.’

Step 16. Close the Internet Explorer window.

Step 17. Repeat Steps 12 thru 15 on OCS Client 1 and OCS Client 2.

Step 18. Open a command prompt window on the OCS Server, and use the command `ping` to verify connection between the OCS Server and the switch, and with both OCS Client 1 and OCS Client 2.

```
ping 192.168.101.160
ping 192.168.101.161
ping 192.168.101.162
```

Repeat these actions on OCS Client 1 and OCS Client 2.

E. INSTALLING THE OCS 10G SOFTWARE ON THE OCS SERVER

The following procedures are designed to install the Oracle Collaboration Suite 10g on a computer (as a Single Computer Installation). The steps of this section follow the installation procedures outlined in Sections 3.4, Starting the Universal Installer, and Section 7.3, Using Advanced Installation for Single-Computer Installation, of the Oracle Collaboration Suite Installation Guide 10g Release 1 (10.1.2). All of the procedures listed in this section are to be completed under the *Administrator* account in Windows.

Step 1. Using a computer connected to the World Wide Web, access the following web site:

http://www.oracle.com/technology/software/products/cs/htdocs/1012winsoft.html.

Step 2: In the center of the webpage, select 'Accept License Agreement.' Download both OCS 10.1.2.2 ZIP files (ocs101220win_1of2.zip and ocs101220win_2of2.zip) to an external hard drive with a USB connection and at least 10 GB in free space (each file is over 1 GB in size).

Step 3. On the OCS Server computer, make a new file folder called 'Oracle' in the Program Files directory on the C: drive (`C:\Program Files\Oracle`). Place both ZIP files into the new 'Oracle' directory (note: the OCS software will not properly install if both files are not located in the same file folder prior to extraction).

Step 4. Extract the ZIP first file to `C:\Program Files\Oracle.` Extract the second ZIP file to the same directory.

Step 5. Install the Java Runtime Environment Version 6, Update 3 plugin on the OCS Server, on OCS Client 1, and on OCS Client 2. Double click the file labeled '`jre-6u3-windows-i586-p-s.exe`' to begin the installation.

Step 6. Find the `setup.exe` file in the `C:\Program Files\Oracle` directory, and double click it. The executable file will check system requirements, display properties, page file size and temp file directory size.

Step 7. Select ‘Advanced Installation’ on the OCS 10g Installer Welcome Page. Then click ‘Next.’

Step 8. In the field labeled ‘Enter the full path of the source directory,’ enter `C:\Program Files\Oracle\Stage\products.xml.` Enter `ocs_home_1` in the Name Field for Destination block. For the full path to the source directory, enter `C:\product\10.1.2\ocs_1.` Click ‘Next.’

Step 9.Select ‘Oracle Collaboration Suite Infrastructure and Applications,’ and click ‘Next.’

Step 10. On the Product-specific Prerequisite Checks page, click ‘Next,’ and the installer verifies system requirements such as memory, disk space, and operating system version.

Step 11. On the Language Selection Screen, select ‘English,’ and click ‘Next.’ Click ‘Next’ again.

Step 12. A page subtitled “Collaboration Suite Infrastructure And Applications Methodology” will load. The following will be included in a caption below the title: ‘These will be installed in the following order and in following locations:… 1. Oracle Collaboration Suite Infrastructure C:\product\10.1.2\OCS_1\infra…. 2. Oracle Collaboration Suite Applications (Middle-tier) c:\product\10.1.2\OCS_1\apps.’ Click “Next”.

Step 13. On the list of Applications Components, uncheck ‘Oracle Voicemail and Fax,’ ‘Voice Conversion Server.’ Leave all the other applications checked, and click ‘Next.’

Step 14. In the Namespace field, enter `dc=cisrlabmlstestbed3, dc=com`. Click ‘Next.’

Step 15. For the OCS 10g database, in the Global Database Name field, enter `orcl1.cisrlabmlstestbed3.com.` The System Identifier (SID) block should be `orcl1`. Set the directory `to C:\product\10.1.2\ocs_1.` Click ‘Next.’

Step 16 Under the captions ‘Specify Database Schema Passwords’ and ‘Specify Application Passwords,’ click the radio button labeled ‘Use the same password for all accounts.’ Set the Database Schema Password to `password123`, and click ‘Next.’

Step 17. Set the Administrative Passwords (for both Infrastructure and Applications) to `password123.` Note: the *Administrator* login for both the Infrastructure and the Application Servers are `ias_admin`. Click ‘Next.’

Step 18. Under the caption “ Specify Oracle Mail Domain Information,” confirm that `cisrlabmlstestbed3.com` is listed as the domain name of the email server, and click ‘Next.’

Step 19. Verify that the default ports for the components to be as follows:

- Oracle Internet Directory Port 389
- Oracle Internet Directory Port (SSL) 636
- Web Cache HTTP Listen 80
- Web Cache HTTP Listen (SSL) 443
- Oracle Mail IMAP4 143
- Oracle Mail IMAP4 Secure 993
- Oracle Mail POP3 110
- Oracle Mail POP3 Secure 995
- Oracle Mail SMTP 25
- Oracle Mail NNTP 119
- Oracle Mail NNTP Secure 563

Click ‘Next.’

Step 20. Under the caption ‘Summary,’ verify selections, and click ‘Install.’ When the focus is set, click ‘Next.’

Step 21. Installer installs the OCS and the Configuration Assistants. When completed, click ‘Exit’ to quit the installer. If an error message pop-up box occurs for lack of MS Office (and it is needed for documentation), click ‘OK,’ and the pop-up goes away. Then try clicking ‘Next’ again.

Step 22. Leave the OCS Server up and running.

Step 23. Proceed to the next section of this Appendix, Section F, Shutting Down the OCS Infrastructure and Application Tiers.

F. SHUTTING DOWN THE OCS INFRASTRUCTURE AND APPLICATION TIERS

The following procedures will be conducted immediately following the installation of the OCS software (as outlined in Section C: Installing the OCS 10g Software on the OCS Server), or when directed to do so (in this Appendix, or in Appendix B: Test Procedures). The OCS shutdown procedures are identical to the steps detailed in Section 2, Stopping and Starting the Oracle Collaboration Suite, of the Oracle Collaboration Suite Administrator’s Guide 10g Release 1 (10.1.2). All of the procedures listed in this section are to be completed under the *Administrator* account in Windows.

Step 1. Open a command prompt window on the OCS Server and shutdown the Oracle Calendar application.

```
cd C:\product\10.1.2\ocs_1\apps\ocas\bin\
ocasctl –stopall
```

Step 2. Shutdown the Oracle Application tier.

```
cd C:\product\10.1.2\ocs_1\apps\opmn\bin\
opmnctl stopall
```

Step 3. Shutdown the Oracle Application Listener process, and shutdown the Application Administrator Console control of the OCS.

```
cd C:\product\10.1.2\ocs_1\apps\bin\
lsnrctl.exe stop listener_es
emctl stop iasconsole
```

Step 4. Shutdown the Infrastructure Administrator Console control of the OCS.

```
cd C:\product\10.1.2\ocs_1\infra\bin\
emctl stop iasconsole
```

Step 5. Shutdown the Infrastructure tier of the OCS.

```
cd C:\product\10.1.2\ocs_1\infra\opmn\bin\
opmnctl stopall
```

Step 6. Shutdown the Oracle Database using SQL commands, and then stop the Infrastructure Listener process.

```
cd C:\product\10.1.2\ocs_1\infra\bin\
sqlplus /nolog
SQL> connect SYS as SYSDBA
Enter password:password123
Connected to an idle instance. (returns on success).
SQL> shutdown immediate
SQL> quit
lsnrctl stop
```

Close the command prompt window.

Step 7. Restart the computer.

G. RESTARTING THE OCS INFRASTRUCTURE AND APPLICATION TIERS

The following procedures will be conducted either (A) after the OCS computer has been restarted following installation of the OCS software (as outlined in Section C: Installing the OCS 10g Software on the OCS Server), (B) when directed to do so (in this Appendix, or in Appendix B: Test Procedures), or (C) anytime the OCS Server has been shutdown, and OCS Infrastructure Tier and the Application Tier needs to be restarted. The OCS restarting procedures are identical to the steps detailed in Section 2, Stopping and Starting the Oracle Collaboration Suite, of the Oracle Collaboration Suite Administrator's Guide 10g Release 1 (10.1.2). All of the procedures listed in this section are to be completed under the *Administrator* account in Windows.

Note: During these steps, if the OCS Server reports that an individual process is already operational, simply note the occurrence, and proceed to the next command line in the procedures.

Step 1. Shutdown the Oracle Infrastructure and Application components using the procedures listed in Section F of this Appendix. If these procedures have already been completed, proceed to Step 2.

Step 2. Open a new command prompt window on the OCS Server to startup the Infrastructure Listener Process in the OCS.

```
cd C:\product\10.1.2\ocs_1\infra\bin\
lsnrctl start
```

Step 3. Startup the Oracle Database using SQL commands.

```
cd C:\product\10.1.2\ocs_1\infra\bin\
sqlplus /nolog
SQL> connect SYS as SYSDBA
Enter password:password123
Connected (returns on success).
SQL> startup
SQL> quit
```

Step 4. Startup the Infrastructure tier in the OCS.

```
cd C:\product\10.1.2\ocs_1\infra\opmn\bin\
opmnctl startall
```

Step 5. Startup the Infrastructure Administrator Console in the OCS.

```
cd C:\product\10.1.2\ocs_1\infra\bin\
emctl start iasconsole
```

Step 6. Close the command prompt window.

Step 7. Open a new command prompt window on the OCS Server, and startup the Application Administrator Console and the Application Listener process of the OCS.

```
cd C:\product\10.1.2\ocs_1\apps\bin\
```

```
emctl start iasconsole
lsnrctl.exe start listener_es
```

Step 8. Startup the Applications tier of the OCS.

```
cd C:\product\10.1.2\ocs_1\apps\opmn\bin\
opmnctl startall
```

Step 9. Startup the Calendar Application of the OCS.

```
cd C:\product\10.1.2\ocs_1\apps\ocas\bin\
ocasctl –start –t ochecklet
ocasctl -start
```

Close the command prompt window.

H. VERIFY THE STATUS OF THE OCS SERVER

Prior to testing, the status of the OCS Application Server, the OCS Infrastructure Server, and the OCS Database will be verified via the command `opmnctl` and also by accessing the OCS Server through either the Internet Explorer web browser or the Mozilla FireFox web browser. The steps of this section follow the server verification procedures outlined in Section 2, Starting and Stopping Oracle Collaboration Suite, of the Oracle Collaboration Suite Administrator's Guide 10g Release 1 (10.1.2). All of the procedures listed in this section are to be completed under the *Administrator* account in Windows.

Step 1. Open a new command prompt window on the OCS Server to verify the status of the OCS Infrastructure components.

```
cd C:\product\10.1.2\ocs_1\infra\opmn\bin
opmnctl status
```

The following infrastructure components should appear as `alive`:

- `HTTP Server`
- `dcm_daemon`
- `OC4J`

- OID

If any of the individual components are shown as being down, manually start up the individual components by using the opmnctl command, in the format of opmnctl startproc ias-component=*component*. For example, to shutdown the HTTP Server component, enter the following command:

```
opmnctl startproc ias-component=HTTP server
```

Refer to Section 2, Starting and Stopping Using opmnctl, of the Oracle Collaboration Suite Administrator's Guide 10g Release 1 (10.1.2), for further guidance on starting and stopping individual processes using the opmnctl command.

Step 2. Open a new command prompt window on the OCS Server to verify the status of the OCS Applications.

```
cd C:\product\10.1.2\ocs_1\apps\opmn\bin
opmnctl status
```

The following applications should appear as alive:

- HTTP Server
- dcm_daemon
- WebCache
- WebCacheAdmin
- OC4J_Portal
- OC4J_OCSA_ADMIN
- OC4J_imeeting
- OC4J_OCSClient
- OC4J_Mail
- Service_Component~
- email_housekeeper
- email_imap
- email_listserver
- email_nntp_in

- email_nntp_out
- email_pop
- email_smtp_in
- email_smtp_out
- email_virus_scrub~
- Node
- OC4J_Content
- Calendar_CSM
- Calendar_CWS
- Calendar_DAS
- Calendar_SNC
- Calendar_ENG
- Calendar_LCK
- rtcpm

If any of the individual components (excluding `logloader` and DSA) are shown as being `down`, manually start up the individual components by using the `opmnctl` command, in the format of `opmnctl startproc ias-component=`*`component`*. For example, to shutdown the `OC4J Mail` component, enter the following command:

```
opmnctl startproc ias-component=OC4J Mail
```

The `OC4J Mail` component will be listed in the second column of the output of the previous `opmnctl status` command. Refer to Section 2, Starting and Stopping Using `opmnctl`, of the Oracle Collaboration Suite Administrator's Guide 10g Release 1 (10.1.2) for further guidance on starting and stopping individual processes using the `opmnctl` command.

Step 3. Verify Internet Explorer 7.0 is configured correctly to access the OCS Server from OCS Client 1 and OCS Client 2. Refer to the steps listed in Appendix A: Installation Procedures, Section D, Settings for Windows and Web Browser Applications for Intranet Testing.

Step 4. On OCS Client 1, double click on the Internet Explorer icon located on the Windows Desktop. In the browser window, type the following address to access the Oracle Collaboration Suite Control Console:

http://ocsserver1.cisrlabmlstestbed3.com:18100

A separate window appears, stating 'The server ocsserver.cisrlabmlstestbed3.com at enterprise-manager is asking for a password.' Enter `ias_admin` for the user name, and `password123` (or whatever has been selected as the `ias_admin` password) for the password.

Step 5. The web browser will display the title 'ORACLE Enterprise Manager 10g Control Console' above the heading 'Farm: orcl.cisrlabmlstestbed3.com.' Further down the page, the following links will be under the heading 'Stand-Alone Instances:'

```
ocsapps.ocsserver1.cisrlabmlstestbed3.com      C:\ product\10.1.2\apps
ocsinfra.ocsserver1.cisrlabmlstestbed3.com     C:\ product\10.1.2\infra
```

Click the link labeled 'ocsapps.ocsserver1.cisrlabmlstestbed3.com.' When prompted, enter `ias_admin` for the username, and `password123` for the password.

Step 6. The web browser will display the title 'ORACLE Enterprise Manager 10g' above the heading 'Application Server Control for Collaboration Suite: ocsapps.ocsserver.cisrlabmlstestbed3.com.' Under the heading 'System Components,' verify the following applications are 'Up' (signified by a green arrow pointing up):

- `Calendar Application System`
- `Calendar Server`
- `Content`
- `Discussions`
- `HTTP Server`
- `Mail Application`
- `OC4J_Content`
- `OC4J_imeeting`
- `OC4J_Mail`

- OC4J_OCSADMIN
- OC4J_OCSClient
- OC4J_Portal
- Real-Time Collaboration
- Search
- Web_Cache
- Workspaces
- Management

If any of the individual applications are shown as being down (red arrow pointed down), refer to Section 2, Starting and Stopping Using the Oracle Collaboration Suite Control Console, of the Oracle Collaboration Suite Administrator's Guide 10g Release 1 (10.1.2) for further guidance on starting and stopping individual processes using the Enterprise Manager Application Server Control Console.

Step 7. Click the link labeled, 'Farm.'

Step 8. The web browser will display the title 'ORACLE Enterprise Manager 10g' above the heading 'Farm: orcl.cisrlabmlstestbed3.com.' Click the link labeled 'ocsinfra.ocsserver.cisrlabmlstestbed3.com.' An additional route to reach this page is achieved by typing the following address in Internet Explorer:

http://ocsserver1.cisrlabmlstestbed3.com:18101

Step 9. The web browser will display the title 'ORACLE Enterprise Manager 10g' above the heading 'Application Server Control for Collaboration Suite: ocsinfra.ocsserver.cisrlabmlstestbed3.com.' Under the heading 'System Components,' verify the following applications are 'Up' (signified by a green arrow pointing up):

- HTTP_Server
- Internet Directory
- OC4J_SECURITY
- Single Sign-On:orasso
- Management

If any of the individual applications are shown as being down (red arrow pointed down), refer to Section 2, Starting and Stopping Using the Oracle Collaboration Suite Control Console, of the Oracle Collaboration Suite Administrator's Guide 10g Release 1 (10.1.2), for further guidance on starting and stopping individual processes using the Enterprise Manager Application Server Control Console.

Step 10. Close the web browser.

Step 11. Open another Internet Explorer web browser window. In the browser window, type the following address to access the Oracle Database Control Console:

`http://ocsserver1.cisrlabmlstestbed3.com:5500/em`

On the page that loads, enter '`SYS`' in the 'User Name' field, enter '`password123`' (or whatever has been selected as the `SYS` password) in the 'Password' field, and select '`Connect as SYSDBA`' in the 'Connect As' drop-down box.

Step 12. The web browser displays the title 'ORACLE Enterprise Manager 10g, Database Control' above the heading 'Database:orcl.ocsserver.cisrlabmlstestbed3.com.' Under the heading 'General,' verify the 'Status' entry is reading 'Up.' An image of a stoplight with a green arrow pointing up will be next to the word 'Status.' **Note:** If this is the first time this web page has been accessed on the server, a page titled 'Oracle 10g Database Licensing Information' will appear. Read the disclosure, and click the button labeled 'Agree' in the bottom right hand corner.

If the 'Status' is reported as 'Down,' refer to Section 6, Starting and Stopping Oracle Collaboration Suite Database, of the Oracle Collaboration Suite Administrator's Guide 10g Release 1 (10.1.2) for further guidance on troubleshooting the Oracle Database.

Step 13. Close the web browser.

I. CREATE TWO OCS USER ACCOUNTS

Create two user accounts in the OCS Database. The steps of this section follow the user account creation procedures outlined in Section 4, Managing Oracle Collaboration Suite Users and Groups: Creating Individual Users, of the Oracle Collaboration Suite Administrator's Guide 10g Release 1 (10.1.2). For instructional purposes, the user account established will be for John Paul Jones. The user account will include All of the procedures listed in this section are to be completed under the *Administrator* account in Windows.

Step 1. On OCS Client 1, double click on the Internet Explorer icon located on the Windows Desktop. In the browser window, type the following address to access the Oracle Provisioning Console:

`http://ocsserver1.cisrlabmlstestbed3.com/oiddas/`

Step 2. The web browser will display the title 'ORACLE Identity Management, Self Service Console.' Click the link labeled 'Login' in the far right corner of the page.

Step 3. In the dark blue field below the title, enter '`orcladmin`' in the 'User Name' field, and enter '`password123`' (or whatever has been selected as the `orcladmin` password) in the 'Password' field. Click the button labeled 'Sign In.'

Step 4. The web browser will display the title 'ORACLE Identity Management: Provisioning Console.' Click the tab labeled 'Directory' on the menu bar.

Step 5. On the next page, under the heading 'Users,' click the tab labeled 'Create.'

Step 6. In the section labeled 'Create User: General,' enter the appropriate information for the new user in the corresponding fields, including 'First Name, Middle Name, Last Name, User ID, Password, Confirm Password', and 'Email Address.' The Email Address will use the following format: `jpjones@cisrlabmlstestbed3.com`. The User Name will correspond to the prefix of the email address (in the previous example, the user name will be `jpjones`.

Note: The email address domain entered for the user account must match **exactly** with the email domain setup in the Oracle Database during installation.

Step 7. Leave the field labeled 'User Default Group' blank, and select 'U.S. Pacific Time Zone' for the field labeled 'Time Zone.' Click the tab labeled 'Next.'

Step 8. On the next page titled 'Create User jpjones: Application Provisioning,' verify that all Components are selected as 'Required,' and click 'Next.'

Step 9. On the next page titled 'Create User jpjones: Application Attributes,' click the tab labeled 'Next.'

Step 10. On the next page titled 'Create User jpjones: Review,' verify that the user information is correct, and click the tab labeled 'Finish.'

Step 11. Click the link labeled 'Logout' in the upper right corner. Close the Browser window. Repeat steps 1 thru 10 to establish a second user account (with a different user ID and email prefix).

Step 11. Click the link labeled 'Logout' in the upper right corner.

J. ACCESSING THE ORACLE COLLABORATION SUITE PORTAL

All of the OCS applications will be accessed from the Oracle Collaboration Suite Portal via the Internet Explorer web browser. All of the procedures listed in this section are to be completed under the *Administrator* account in Windows.

Step 1. On OCS Client 1, double click on the Internet Explorer icon located on the Windows Desktop. In the browser window, type the following address to access the Oracle Collaboration Suite Portal:

http://ocsserver1.cisrlabmlstestbed3.com:80

Step 2. The greeting 'Welcome to Oracle Collaboration Suite' will appear under the title 'ORACLE Collaboration Suite.' Under the Heading 'End-User Resources,' click the link titled 'Collaboration Suite Portal.' **Note:** The Collaboration Suite Portal page is

known also known as the Single Sign-On (SSO) Page in the Oracle Collaboration Suite Administrator's Guide 10g Release 1 (10.1.2).

Step 3. The web browser will display the title 'ORACLE Collaboration Suite.' In the dark blue field below the title, enter the username of the first Oracle User Account established in Section B (Create Two OCS User Accounts) in the 'User Name' field, and enter the corresponding password in the 'Password' field. Click the button labeled 'Sign In.'

Step 4. The browser will display the Oracle Collaboration Suite User Portal. The default portal will be divided graphically into the following sections, from the top left of the page, going clockwise: `Links, News, Mail, Content Services, Web Conferencing, Tasks, and Calendar`.

Step 5. Keep the browser window open on OCS Client 1.

Step 6. Repeat steps 1 thru 4 on OCS Client 2, using the second Oracle User Account established in Section I to access the Oracle Collaboration Suite Portal.

Step 7. Keep the browser window open on OCS Client 2.

Step 8. If applicable, test the OCS Applications, using the test procedures outlined in Appendix B, Test Procedures.

Step 9. When finished, click the link labeled 'Logout' in the upper right corner of the browser window to logout. Close the browser window.

K. CONNECTING THE OCS SERVER TO THE SIMULATED MLS ENVIRONMENT

The installation steps listed in this section outline the steps to connect the OCS Server to the same switch where the XTS-400 server resides. See Figure 2, Network Topology for Direct Connection Testing, in this Appendix, for a description of what the network connections should look like prior to executing these steps.

Step 1. Shutdown the OCS Infrastructure tier and the OCS Application tier by repeating the procedures listed in Section G, Shutting Down the OCS Infrastructure and Application Tiers, in this Appendix.

Step 2. Log off the OCS Server using the Windows *Administrator* accounts. Shutdown the OCS Server.

Step 3. Disconnect the Ethernet cable from the OCS Server to Switch 1.

Step 4. Reconnect the Ethernet cable from the OCS Server to Switch 2 (the switch where the XTS-400 server resides).

Step 5. Verify that the changes made in Steps 1 thru 4 resemble the topology illustrated in Figure C, Network Topology for Simulated MLS Testing, below.

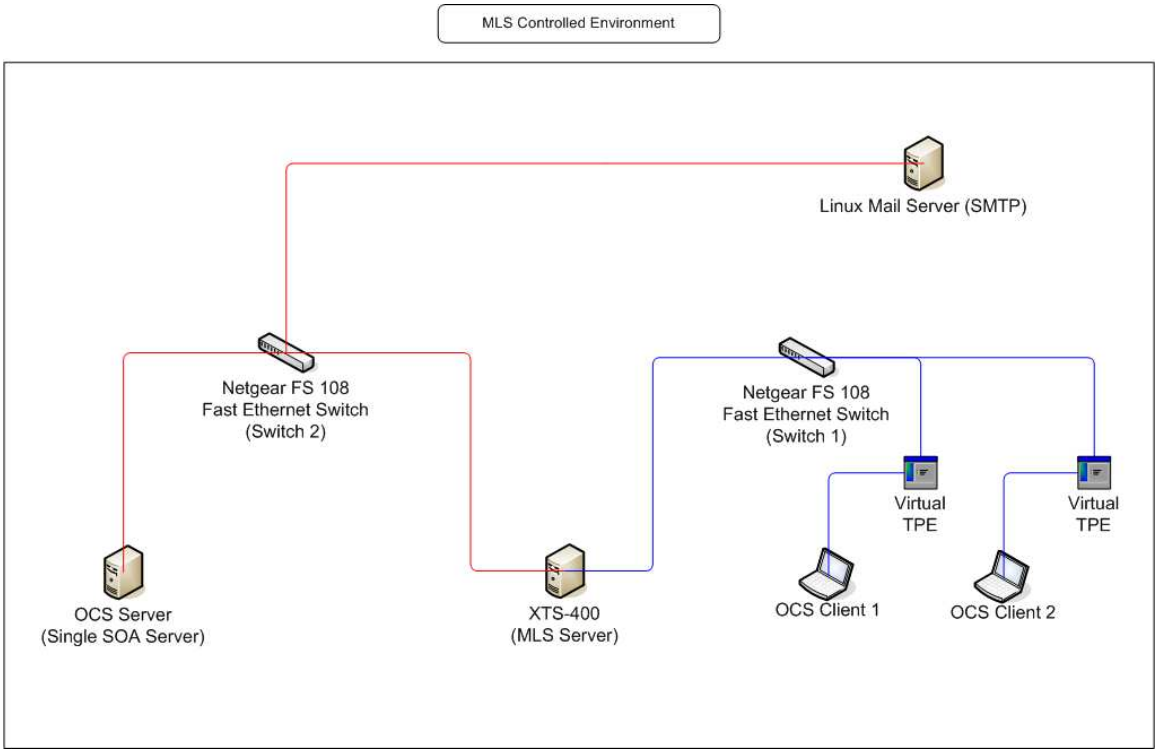

Figure C. Network Topology for Simulated MLS Testing.

Step 6. Turn on the OCS Server. Login to Windows using the Windows *Administrator* account.

Step 7. Startup the OCS Infrastructure tier and the OCS Application tier by repeating the procedures listed in Section H, Starting Up the OCS Infrastructure and Application Tiers, in this Appendix.

L. SETTINGS FOR WINDOWS AND WEB BROWSER APPLICATIONS FOR MLS TESTING

Prior to conducting the MLS testing detailed in Chapter IV, several modifications will be completed on the OCS Server and on the OCS Clients. These changes include rewriting the `hosts` file (OCS Clients only), changing the IP address associated with each machine (OCS Clients only), modifying IP addresses (OCS Clients only), turning the Windows Firewall off, and altering the settings on the Mozilla FireFox web browser. All of the procedures listed in this section are to be completed under the *Administrator* account in Windows.

Step 1. Rewrite the IP addresses and domain names listed in the hosts.ini file on OCS Client 1 and on OCS Client 2. Login as the *Administrator* on the OCS Client 1. Open the `hosts` file located in the `C:\\WINDOWS\system32\drivers\etc\` directory. Replace the existing text with the following:

```
127.0.0.1         localhost.localdomain                localhost
192.168.0.31      ocsclient1.cisrlabmlstestbed1.com    ocsclient1
192.168.0.32      ocsclient2.cisrlabmlstestbed1.com    ocsclient2
192.168.101.164   ocsserver1.cisrlabmlstestbed3.com    ocsserver1
192.168.101.160   ocsserver.cisrlabmlstestbed3.com     ocsserver
192.168.0.130     mlsserver.cisrlabmlstestbed1.com     mlsserver
```

Repeat these actions on OCS Client 2.

Step 2. On the OCS Server, open the Network Connections tab in the Windows Control Panel. Right click the Local Area Connections icon, and select ‘Properties.’ In the window that appears, click on the ‘Internet Protocol (TCP/IP)’ icon, and click on the

‘Properties’ tab. Select ‘Use the Following IP Address,’ and depending on which computer is being modified, set the IP address as follows:

- OCS Server (ORACLE) `192.168.101.164`
- OCS Client 1 `192.168.0.31`
- OCS Client 2 `192.168.0.32`

In the ‘Subnet Mask’ field, enter `255.255.255.0.` In the ‘Default Gateway’ field, enter `192.168.0.30`, which is the XTS-400’s Proxy server address. Repeat these actions on OCS Client 1 and OCS Client 2.

Step 3. Open the Control Panel on the OCS Server. Double click the ‘System Properties’ icon, and then click ‘Computer Name.’ Change the ‘Name’ field to `OCSSERVER1`. Make sure the ‘Workgroup’ radio button under “Member Of:” is selected, and that the ‘Workgroup’ field includes the title `WORKGROUP`. Above the ‘Member Of:’ area, click “More,” and enter `cisrlabmlstestbed3.com` in the Domain field. Restart the computer when prompted.

Step 4. Open the Control Panel on the OCS Client 1. Double click the ‘System Properties’ icon, and then click ‘Computer Name.’ Change the ‘Name’ field to `OCSCLIENT1`. Make sure the ‘Workgroup’ radio button under “Member Of:” is selected, and that the ‘Workgroup’ field includes the title `WORKGROUP`. Above the ‘Member Of:’ area, click “More,” and enter `cisrlabmlstestbed1.com` in the Domain field. Restart the computer when prompted.

Step 5. Open the Control Panel on the OCS Client 2. Double click the ‘System Properties’ icon, and then click ‘Computer Name.’ Change the ‘Name’ field to `OCSCLIENT2`. Make sure the ‘Workgroup’ radio button under “Member Of:” is selected, and that the ‘Workgroup’ field includes the title `WORKGROUP`. Above the ‘Member Of:’ area, click “More,” and enter `cisrlabmlstestbed1.com` in the Domain field. Restart the computer when prompted.

Step 6. On the OCS Server, open the Network Connections tab in the Windows Control Panel. Right click the Local Area Connections icon, and select ‘Properties.’ In the window that appears, click on the ‘Advanced’ tab. In the ‘Windows Firewall’ area, click the ‘Settings’ tab and disable the Windows Firewall/ICS Service.

Step 7. On OCS Client 1, open the Network Connections tab in the Windows Control Panel. Right click the Local Area Connections icon, and select ‘Properties.’ In the window that appears, click on the ‘Advanced’ tab. In the ‘Windows Firewall’ area, click the ‘Settings’ tab. In the window that appears, select ‘Off (not recommended)’ and click ‘OK.’ Restart the computer when prompted. Repeat these steps for OCS Client 2.

Step 8. On the OCS Server, double click on the Mozilla FireFox icon on the desktop. On the FireFox menu bar, click ‘Tools,’ and select ‘Options.’ Under the ‘Advanced’ section of the ‘Options’ bar, open the ‘Network’ tab. Click ‘Settings,’ and select ‘Manual Proxy Configuration.’ In the ‘IP address’ field, enter `192.168.0.130`, and for ‘Port’ enter `80.` Check the box labeled ‘Use Proxy for all Protocols.’ Click ‘OK.’ Click the ‘Content’ tab on the ‘Options’ bar, and ensure that both ‘Enable Java’ and ‘Enable Javascript’ are checked. Close the FireFox window.

Step 9. Repeat Step 8 on OCS Client 1, and OCS Client 2.

Step 10. On the OCS Server, double click on the Internet Explorer 7.0 icon on the desktop. On the Explorer menu bar, click ‘Tools,’ and select ‘Internet Options.’ On the ‘Intenet Options’ tab, click ‘Privacy.’ In the ‘Privacy’ area, uncheck ‘Turn Pop-up Blocker On,’ and set the ‘Settings’ bar to ‘Accept All Cookies.’

Step 11. On the ‘Intenet Options’ tab, click ‘Advanced,’ and ensure all of the following boxes remain **unchecked**:

- Enable Integrated Windows Authentication
- Enable interactive XML HTTP support
- HTTP 1.1
- HTTP 1.1 thru Proxy

- SSL 2.0
- Check for valid signatures on Downloaded Programs

Step 12. Open the 'Network' tab, and select 'Manual Proxy Configuration.' In the 'IP address' field, enter `192.168.0.130`, and for 'Port' enter `80`. Check the box labeled 'Use Proxy for all Protocols.' Click the 'Content' tab on the 'Options' bar, and ensure that both 'Enable Java' and 'Enable Javascript' are checked.

Step 13. Close the Internet Explorer window.

Step 14. Repeat Steps 10 thru 13 on OCS Client 1 and OCS Client 2.

Step 15. Open a command prompt window on OCS Client 1, and use the command `ping` to verify connection between the OCS Clients and the proxy server (the XTS-400).

```
ping 192.168.0.31
ping 192.168.0.32
ping 192.168.0.130
```

Repeat these actions on OCS Client 2.

M. ESTABLISHING A SINGLE LEVEL CONNECTION WITH THE MLS SERVER

A simulated single level connection in the untrusted environment will be established prior to testing the OCS in the simulated MYSEA multilevel testbed. If testing is being conducted on the OCS Server directly connected to the clients in an isolated intranet (as described in Section B), ignore this section. This testing used an XTS-400 server running the STOP 6.1 operating system.

Step 1. Turn on the XTS-400 server.

Step 2. Following the line 'Enter Partition Number,' type `2` and hit Enter.

Step 3. Login to the MLS server.

```
Enter user name?                                    admin
Enter password?                                     xts400
```

Step 4. Startup all of the daemons that are part of the MYSEA testbed.

```
SAK
Enter command?                                          sl
Enter new session security level and categories?        max
Enter new session integrity level and categories?       max
Is the level correct?                                   y
SAK
Enter command?                                          startup
```

Step 5. Change the security level to ‘min’ and the integrity level to ‘oss.’.

```
SAK
Enter command?                                          sl
Enter new session security level and categories?        min
Enter new session integrity level and categories?       oss
Is the level correct?                                   y
SAK
Enter command?                                          run
```

Step 6. Upon the completion of testing, logout of the MLS server.

```
SAK
Enter command?                                          shutdown
```

Step 7. Shut down the MLS server.

N. CONNECTING THE OCS CLIENTS TO A SINGLE LEVEL SESSION VIA THE VIRTUAL TRUSTED PATH EXTENSION DEVICES

To access a single level session on the MLS server, both OCS Client 1 and OCS Client 2 are configured to run a virtual trusted path extension (TPE). This virtual TPE consists of an executable program designed to provide a trusted path between the OCS Client and the MLS server. All of the procedures listed in this section are to be completed under the *Administrator* account in Windows.

Step 1. On the OCS Client 1 desktop, double click the Virtual TCBE file labeled ‘TCBE.exe.’

Step 2. In the Java window that appears, verify the Server IP Address field to be `192.168.0.130`.

Step 3. Click the large red ‘SAR’ tab in the Java window.

Step 4. Login to the MLS server as `mdemo1.`

```
Enter user name:                                    mdemo1
Enter password:                                     tcxuser
```

Step 5. Click the ‘SAR’ tab.

Step 6. Initiate a `SIM_SECRET` session.

```
Enter command:                                      sl
Enter session level?                                SIM_SECRET
```

Step 7. Click the ‘SAR’ tab.

Step 8. Run the session.

```
Enter command:                                      run
```

Step 9. Upon the completion of testing, logout and close the connection.

```
SAK
Enter command?                                      logout
```

Step 10. Close the Java window.

Step 11. Repeat steps 1 thru 10 on OCS Client 2, logging in as `mdemo2` and using the same password (`tcxuser`).

O. VERIFY THE STATUS OF THE OCS SERVER (MLS TESTING)

Prior to MLS testing, a single level connection to the OCS Server will be verified using a virtual TPE and either the Mozilla FireFox web browser or the Internet Explorer web browser.

Step 1. Establish a single level connection with the simulated MLS server by repeating the procedures in Section M: Establishing a Single Level Connection with the MLS server, of this appendix. If this has already been accomplished, skip to Step 2.

Step 2. Repeat the procedures in Section N: Connect the OCS Clients to a single level Session via the Virtual Trusted Path Extension Devices, of this appendix. If this has already been accomplished, skip to Step 3

Step 3. Repeat the procedures listed in Section H, Verify the Status of the OCS Server, in this Appendix.

Step 4. Proceed to Appendix B: Test Procedures, to test the OCS Applications while the OCS Server is connected to the MLS Server.

P. CONFIGURING THE SMTP SETTINGS ON THE OCS SERVER

The instructions for configuring the Simple Mail Transfer Protocol settings on the OCS Server are included in Appendix B: Test Procedures.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B: TEST PROCEDURES

This appendix describes procedures for testing the following applications of the Oracle Collaboration Suite (OCS) 10g, version 10.1.2, in the Monterey Security Architecture (MYSEA) simulated multilevel testbed:

- Oracle Web Mail (web browser)
- Oracle Content Services (web browser)
- Oracle Real-Time Collaboration: Web Conferencing (web browser)
- Oracle Calendar (web browser)
- Oracle Workspaces (web browser)
- Oracle Discussions (web browser)
- Oracle Real-Time Collaboration: RTC Instant Messenger (rich media client)
- Oracle Content Services: 'Oracle Drive' (rich media client)
- SMTP mail exchange (with Linux Server)

The goal of this appendix is to provide a set of minimal tests (one set of tests per Oracle application) capable of verifying the functionality of the nine Oracle applications listed above. This appendix is divided into individual sections corresponding to the OCS applications listed above. Aside from Section F (Oracle Discussions), none of the sections are dependent on each other (an Oracle Workspace should be created (as described in Section E) prior to conducting the procedures in Section F). Each section can be completed individually, in no particular sequence. For a further description of these applications, refer to Chapter II of this thesis, and also Section 6, Installing Oracle Collaboration Suite 10g Applications, of the Oracle Collaboration Suite (OCS) Installation Guide 10g Release 1 (10.1.2) [14].

As discussed in Appendix A, testing the OCS 10g applications will occur in two phases: (1) testing the OCS applications with the OCS Server directly connected to the OCS clients, and (2) testing the OCS applications at the single level using the XTS-400 server as a proxy. Prior to testing these applications, the following steps and procedures

listed in Appendix A (Installation Procedures) should be completed, depending on which phase of testing is being currently administered: (a) for Phase 1 testing (Direct Connection Testing), complete sections A through J in Appendix A, and (b) for Phase 2 testing (Single Level Testing, Using XTS-400 as a Proxy), complete all of Appendix A (Sections A through P) prior to testing the applications for Phase 2.

A. TEST THE ORACLE WEB MAIL APPLICATION (WEB BROWSER)

The Oracle Web Mail (web browser) application will be accessed from the Oracle Collaboration Suite Portal using one of two web browsers: either the Internet Explorer web browser, or the Mozilla FireFox web browser. These procedures can be completed on either OCS Client 1 or OCS Client 2. All of the procedures listed in this section are to be completed under the *Administrator* account in Windows.

The procedures in this section verify that the Oracle Mail application is functioning properly on the OCS 10g server. The three parts of this section of the Appendix are based on three individual tests: (1) the Web Browser test, (2) the Oracle User Login test, and (3) the Oracle Web Mail test. The tests in this section should be completed in order, from the first to the last. If any one of the three tests is marked as a 'Fail,' it will be recorded that the Oracle Mail application did not function as expected.

1. Web Browser Test

This test will verify that the web browser can (1) open correctly, and (2) access the web site http://ocsserver1.cisrlabmlstestbed3.com:80. Individual subtests (ex. Test 1, Test 2.) will be marked 'P' for Pass, or 'F' for Fail. If one (or more) of the subtests of this test is marked as a Fail, the test will be recorded as a 'Fail' overall. Table 3 describes the details of this test procedure.

Table 3. Web Browser Test

Web Browser Tests: To verify that the web browser can (1) open correctly, and (2) access the web site http://ocsserver1.cisrlabmlstestbed3.com:80. Individual subtests (ex. Subtest 1, Subtest 2) will be marked 'P' for Pass, or 'F' for Fail.				
1. Subtest 1: Open the web browser.				
	A.	Open the Web Browser. The resultant page should be blank, or if not configured, then the browser will stall, and eventually time out, stating, "Web Site Could Not Be Found."		
	PASS criteria for Oracle Web Browser Subtest 1: The web browser will open and will state "Web Site Could Not Be Found" (unless it has been configured for the Oracle Collaboration Suite Portal).		**P/F**	
2. Subtest 2: Access the Oracle Collaboration Suite Portal using the web browser				
	A.	Access the web site http://ocsserver1.cisrlabmlstestbed3.com:80. The browser window should change to the heading 'End-User Resources. The greeting 'Welcome to Oracle Collaboration Suite' will appear under the title 'ORACLE Collaboration Suite.'		
	PASS criteria for Oracle Web Browser Subtest 2: The Oracle Collaboration Suite Portal will appear in the browser as detailed in block A.		**P/F**	
Overall Pass/Fail? All Web Browser Subtests must be marked with a 'Pass.'			**P/F**	

2. Oracle User Login Test

This test will verify that that an Oracle User can login to the Oracle Collaboration Suite User Portal via the web browser. The individual subtest (Test 1) will be marked 'P' for Pass, or 'F' for Fail. If the individual subtest of this test is marked as a Fail, the test will be recorded as a Fail overall. Table 4 describes the details of this test procedure.

Table 4. Oracle User Login Test

Oracle User Login Test: To verify that an Oracle User can login to the Oracle Collaboration Suite User Portal via web browser. Individual tests (ex. Subtest 1, Subtest 2) will be marked 'P' for Pass, or 'F' for Fail.				
1. Subtest 1: Access and login to the Oracle Collaboration Suite User Portal using the web browser.				
	A.	Access the web site http://ocsserver1.cisrlabmlstestbed3.com:80. The browser window should change to the heading 'End-User Resources. The greeting 'Welcome to Oracle Collaboration Suite' will appear under the title 'ORACLE Collaboration Suite.'		
	B.	Click the link titled "Collaboration Suite Portal," which lies one-third down the page in the center. The browser window that opens should be titled "Collaboration Suite Portal."		
	C.	Enter the user name of the first Oracle User Account in the dark blue field below the title, and enter the password `password123`. Press return (or click 'Go' on the browser). The browser will open to the Oracle Collaboration Suite User Portal. **Note:** If the Oracle User Account is already logged in, the window will automatically redirect to the User Portal (proceed to step D).		
	D.	The Oracle Collaboration Suite User Portal will be divided graphically into the following sections: `Links`, `News`, `Mail`, `Content Services`, `Web Conferencing`, `Tasks`, and `Calendar`. Hit the 'Refresh' button on the browser if any of the sections are listed as 'Unavailable.'		
	PASS criteria for Oracle User Login Subtest 1: The Oracle Collaboration Suite User Portal can be accessed by the first Oracle User Account, and correctly displays all of the sections listed in part D.		**P/F**	
Overall Pass/Fail? The Oracle User Login Subtest must be marked with a 'Pass.'			**P/F**	

3. Oracle Web Mail Test

This test will verify that the Oracle Web Mail (web browser) application can (a) open Web Mail from the User Portal, and (b) send another Oracle user an email and verify that the message was sent. Individual subtests (ex. Subtest 1, Subtest 2) will be marked 'P' for Pass, or 'F' for Fail. If one (or more) of the subtests of this test is marked as a Fail, the test will be recorded as a 'Fail' overall. Table 5 describes the details of this test procedure.

Table 5. Oracle Web Mail Test

Oracle Web Mail Test: To verify that the Oracle Web Mail web browser application can (a) open Web Mail from the User Portal, and (b) send another Oracle user an email and verify that the message was sent. Individual tests (ex. Subtest 1, Subtest 2) will be marked 'P' for Pass, or 'F' for Fail.				
1. Subtest 1: Access the Oracle Web Mail Application from the Oracle User Portal.				
	A.	Login to the Oracle Collaboration Suite User Portal (as demonstrated in Test 1 of the Oracle User Login Test). The Oracle Collaboration Suite User Portal will be divided graphically into the following sections: `Links`, `News`, `Mail`, `Content Services`, `Web Conferencing`, `Tasks`, and `Calendar`. Hit the 'Refresh' button on the browser if any of the sections are listed as 'Unavailable.'		
	B.	Click on the link labeled `Mail` (which is below `News` and above `Content Services`). A new browser window (the Web Mail window) should appear with an image of an envelope and the word `MAIL` immediately to the right.		
	PASS criteria for Oracle Web Mail Subtest 1: An additional window (the Web Mail window) will appear, displaying the content described in Block B.		**P/F**	
2. Subtest 2: Send another Oracle User an email using Oracle Web Mail				
	A.	In the Web Mail Window (see Step 1.B. above), the following headings will appear above the `MAIL` image, from left to right: `New, View, Go, Actions, Print, Delete, and Find People.` Click the heading labeled `New`, and a new browser window will appear (the New Message Window).		
	B.	In the New Message Window, click the heading labeled `Format`, and on the drop down box that appears, make sure that `HTML` is selected.		
	C.	Click the text field to the right of the `To:` button, and type in the email address of the second Oracle User Account (e.g., cmgilkey@cisrlabmlstestbed3.com).		
	D.	Click the text field to the right of the `Subject` heading, and type `Email Test.`		
	E.	Click the large text field immediately below `Subject` and type `test` Click the `Send` button (the New Message window will close).		
	F.	On the Web Mail Window, click the `Sent Items` heading. The large title to the right should now read `Sent Items.` The field below this title should contain the email that was sent in Step D (titled `Email Test`).		
	PASS criteria for Oracle Web Mail Subtest 2: The Oracle Webmail constructed in Steps A through E will appear in the `Sent Items` folder (as noted in Step F).		**P/F**	
Overall Pass/Fail? All Oracle Web Mail Subtests must be marked with a 'Pass.'			**P/F**	

B. TEST THE ORACLE CONTENT SERVICES (WEB BROWSER) APPLICATION

The Oracle Content Services (web browser) application will be accessed from the Oracle Collaboration Suite Portal using one of two web browsers: either the Internet Explorer web browser, or the Mozilla FireFox web browser. These procedures can be completed on either OCS Client 1 or OCS Client 2. All of the procedures listed in this section are to be completed under the *Administrator* account in Windows. Ensure that the pop-up blockers on Internet Explorer and Mozilla FireFox are disabled. A single Portable Network Graphics (PNG) file of the user's choosing should be saved to the Windows Desktop file Directory on both OCS Client 1 and OCS Client 2 prior to testing the Oracle Content Services Application.

The procedures in this section verify that the Oracle Content Services Application is functioning properly on the OCS 10g server. The three parts of this section of the Appendix are based on three individual tests: (1) the Web Browser test, (2) the Oracle User Login test, and (3) the Oracle Content Services (web browser) test. The tests in this section should be completed in order, from the first to the last. If any one of the three tests is marked as a 'Fail,' it will be recorded that the Oracle Content Services (web browser) application did not function as expected.

1. Web Browser Test

This test will verify that the web browser can (1) open correctly, and (2) access the web site http://ocsserver1.cisrlabmlstestbed3.com:80. Individual subtests (ex. Subtest 1, Subtest 2) will be marked 'P' for Pass, or 'F' for Fail. If one (or more) of the subtests of this test is marked as a Fail, the test will be recorded as a 'Fail' overall. Table 3 describes the details of this test procedure.

2. Oracle User Login Test

This test will verify that that an Oracle User can login to the Oracle Collaboration Suite User Portal via the web browser. The individual subtest (Subtest 1) will be marked

'P' for Pass, or 'F' for Fail. If the individual subtest of this test is marked as a Fail, the test will be recorded as a Fail overall. Table 4 describes the details of this test procedure.

3. Oracle Content Services Application (Web Browser) Test

This test will verify that the Oracle Content Services (web browser) application can (a) open Content Services from the User Portal, and (b) upload a file from the client's desktop file directory into the Oracle User folder corresponding to the Oracle User logged in. Individual subtests (ex. Subtest 1, Subtest 2.) will be marked 'P' for Pass, or 'F' for Fail. If one (or more) of the subtests of this test is marked as a Fail, the test will be recorded as a ‘Fail’ overall. Table 6 describes the details of this test procedure.

Table 6. Oracle Content Services (Web Browser) Test

Oracle Content Services (Web Browser) Tests: To verify that the Oracle Content Services web browser application can (a) open Content Services from the User Portal, and (b) upload a file from the client's desktop file directory into the Oracle User folder corresponding to the Oracle User. Individual subtests (ex. Subtest 1, Subtest 2) will be marked 'P' for Pass, or 'F' for Fail.				
1. Subtest 1: Access the Oracle Content Services Application from the Oracle User Portal.				
	A.	Login to the Oracle Collaboration Suite User Portal (as demonstrated in Test 1 of the Oracle User Login Test). The Oracle Collaboration Suite User Portal will be divided graphically into the following sections: `Links`, `News`, `Mail`, `Content Services`, `Web Conferencing`, `Tasks`, and `Calendar` Hit the 'Refresh' button on the browser if any of the sections are listed as 'Unavailable.'		
	B.	Click on the link labeled `Content Services` (which is below `Mail`). A new browser window (the Content Services window) should appear with the an image of an envelope directly under the heading `Current Location`.		
	PASS criteria for Oracle Content Services Subtest 1: An additional window (the Content Services window) will appear, displaying the content described in Block B.		**P/F**	
2. Subtest 2: Upload a file from the client's desktop file directory into the Oracle User folder.				
	A.	In the Content Services Window (see Step 1.B. above), beneath the `Current Location` header will be one folders labeled `cisrlabmlstestbed3.com` and one below that labeled `users`. Double-click the `users` folder.		
	B.	A list of Oracle User directories will appear below the `users` folder. Double-click the user directory corresponding to the first Oracle User Account.		
	C.	The large text field to the right of the folders will contain a folder labeled `Trash`, and any other files that have been uploaded into the user directory of the first Oracle User Account. Right click somewhere in the text field (other than over the `Trash` folder), and a new window (the File Directory window) will appear.		
	D.	In the File Directory window, there will be a row of identical buttons labeled `Browse`. Click the topmost `Browse` button. A new window will appear.		
	E.	In the File Upload window, click the icon labeled `Desktop`. Left click on the Portable Network Graphics (PNG) file that was saved to the Windows Desktop file directory. When the PNG file name appears in the text field labeled `File name:`, click the button labeled `Open`. The File Upload Window will close.		
	F.	In the File Directory window, the directory location of the PNG file should now appear in the `File` text field immediately to the left of the topmost `Browse` button. Click the `Upload` button. The File Directory Window will close.		
	G.	In the Content Search window, the PNG file appears under the `Trash` folder.		
	PASS criteria for Oracle Content Services Subtest 2: The file uploaded in Steps A through F will appear in the Oracle User's `User` folder (as noted in Step G).		**P/F**	
Overall Pass/Fail? All Oracle Content Services (Web Browser) Subtests must be marked with a 'Pass.'			**P/F**	

C. TEST THE ORACLE RTC WEB CONFERENCING (WEB BROWSER) APPLICATION

The Oracle RTC Web Conferencing (web browser) application will be accessed from the Oracle Collaboration Suite Portal using one of two web browsers: either the Internet Explorer web browser, or the Mozilla FireFox web browser. These procedures can be completed on either OCS Client 1 or OCS Client 2. All of the procedures listed in this section are to be completed under the *Administrator* account in Windows. Ensure that the pop-up blockers on Internet Explorer and Mozilla FireFox are disabled.

The procedures in this section verify that the Oracle RTC Web Conferencing application is functioning properly on the OCS 10g server. The three parts of this section of the Appendix are based on three individual tests: (1) the Web Browser test, (2) the Oracle User Login test, and (3) the Oracle RTC Web Conferencing (web browser) test. The tests in this section should be completed in order, from the first to the last. If any one of the three tests is marked as a 'Fail,' it will be recorded that the Oracle RTC Web Conferencing (web browser) application did not function as expected.

1. Web Browser Test

This test will verify that the web browser can (1) open correctly, and (2) access the web site http://ocsserver1.cisrlabmlstestbed3.com:80. Individual subtests (ex. Subtest 1, Subtest 2) will be marked 'P' for Pass, or 'F' for Fail. If one (or more) of the subtests of this test is marked as a Fail, the test will be recorded as a 'Fail' overall. Table 3 describes the details of this test procedure.

2. Oracle User Login Test

This test will verify that that an Oracle User can login to the Oracle Collaboration Suite User Portal via the web browser. The individual subtest (Subtest 1) will be marked 'P' for Pass, or 'F' for Fail. If the individual subtest of this test is marked as a Fail, the test will be recorded as a Fail overall. Table 4 describes the details of this test procedure.

3. Oracle RTC Web Conferencing Application (Web Browser) Test

This test will verify that the Oracle RTC Web Conferencing (web browser) application can (a) be accessed from the User Portal, and (b) create an instant web conference. Individual subtests (ex. Subtest 1, Subtest 2.) will be marked 'P' for Pass, or 'F' for Fail. If one (or more) of the subtests of this test is marked as a Fail, the test will be recorded as a ‘Fail’ overall. Table 7 describes the details of this test procedure.

Table 7. Oracle RTC Web Conferencing Tests

Oracle Real-Time Collaboration: Web Conferencing (Web Browser) Tests: To verify that the Oracle Real-Time Collaboration (RTC) Web Conferencing web browser application can (a) be accessed from the User Portal, and (b) create an instant web conference. Individual subtests (ex. Subtest 1, Subtest 2) will be marked 'P' for Pass, or 'F' for Fail.				
1. Subtest 1: Access the Oracle RTC Web Conferencing Application from the Oracle User Portal.				
	A.	Login to the Oracle Collaboration Suite User Portal (as demonstrated in Test 1 of the Oracle User Login Test). The Oracle Collaboration Suite User Portal will be divided graphically into the following sections: `Links`, `News`, `Mail`, `Content Services`, `Web Conferencing`, `Tasks`, and `Calendar` Hit the 'Refresh' button on the browser if any of the sections are listed as 'Unavailable.'		
	B.	Click on the link labeled `Web Conferencing` (which is below `Mail`). A new browser window (the RTC window) should appear with the text `Oracle Collaboration Suite Real-Time Collaboration` across the top of the page.		
	PASS criteria for Oracle Real-Time Collaboration Web Conferencing Test 1: An additional window (the RTC window) appears, with content described in Block B.		**P/F**	
2. Subtest 2: Create an Instant Web Conference				
	A.	In the RTC Window (see Step 1.B. above), the title `Oracle Collaboration Suite Real-Time Collaboration` will appear across the top of the window. On the right side of the window will be 3 gray boxes. In the topmost gray box (with the header `Instant Conference`) click the text field to the right of the words `Conference Title`, and type `Test 1`. Click the `Start Conference` button in the `Instant Conference` box. The RTC Window will load a new page.		
	B.	The RTC Window will display a new page with the heading `Console Initialization in Progress` at the top left of the page. Once the initialization is complete, a large Oracle RTC Web Conference Interface will appear above the RTC Window. **Note:** Mozilla FireFox users may receive a warning stating 'Compatibility Issues Found: Potential for limited feature support.' If this occurs, click `continue`, and the initialization will finish.		
	C.	A second window (Oracle Conference Details) will appear. Click `Apply`.		
	D.	In the middle of the Oracle RTC Web Conference Interface, locate a chat bubble icon to the right of a text field with the Oracle User's name (ex. `jpjones`). . Click the icon. The RTC Web Conference Interface will include a chat interface.		
	E.	By default, the cursor relocates in a text field to the right of the chat interface. Type `The Quick Brown Fox Jumps Over the Lazy Dog`. Hit Enter.		
	F.	Verify that the text entered is displayed in the middle of chat interface.		
	PASS criteria for Oracle Web Conference Subtest 2: An Instant Web Conference can be created, and it can display chat responses.		**P/F**	
Overall Pass/Fail? All Oracle Real-Time Collaboration: Web Conference (Web Browser) Subtests must be marked with a 'Pass.'			**P/F**	

D. TEST THE ORACLE CALENDAR (WEB BROWSER) APPLICATION

The Oracle Calendar (web browser) application will be accessed from the Oracle Collaboration Suite Portal using one of two web browsers: either the Internet Explorer web browser, or the Mozilla FireFox web browser. These procedures can be completed on either OCS Client 1 or OCS Client 2. All of the procedures listed in this section are to be completed under the *Administrator* account in Windows. Ensure that the pop-up blockers on Internet Explorer and Mozilla FireFox are disabled.

The procedures in this section verify that the Oracle Calendar (web browser) application is functioning properly on the OCS 10g server. The three parts of this section of the Appendix are based on three individual tests: (1) the Web Browser test, (2) the Oracle User Login test, and (3) the Oracle Calendar (web browser) test. The tests in this section should be completed in order, from the first to the last. If any one of the three tests is marked as a 'Fail,' it will be recorded that the Oracle Calendar (web browser) application did not function as expected.

1. Web Browser Test

This test will verify that the web browser can (1) open correctly, and (2) access the web site http://ocsserver1.cisrlabmlstestbed3.com:80. Individual subtests (ex. Subtest 1, Subtest 2) will be marked 'P' for Pass, or 'F' for Fail. If one (or more) of the subtests of this test is marked as a Fail, the test will be recorded as a 'Fail' overall. Table 3 describes the details of this test procedure.

2. Oracle User Login Test

This test will verify that that an Oracle User can login to the Oracle Collaboration Suite User Portal via the web browser. The individual subtest (Subtest 1) will be marked 'P' for Pass, or 'F' for Fail. If the individual subtest of this test is marked as a Fail, the test will be recorded as a Fail overall. Table 4 describes the details of this test procedure.

3. Oracle Calendar Application (Web Browser) Test

This test will verify that the Oracle Calendar (web browser) application can (a) be accessed from the User Portal, and (b) create an appointment on the user's calendar. Individual subtests (ex. Subtest 1, Subtest 2.) will be marked 'P' for Pass, or 'F' for Fail. If one (or more) of the subtests of this test is marked as a Fail, the test will be recorded as a 'Fail' overall. Table 8 describes the details of this test procedure.

Table 8. Oracle Calendar Test

Oracle Calendar (Web Browser) Tests: To verify that the Oracle Calendar web browser application can (a) be opened from the User Portal, and (b) set an appointment (meeting) on the user's calendar. Individual subtests (ex. Subtest 1, Subtest 2) will be marked 'P' for Pass, or 'F' for Fail.				
1. Subtest 1: Access the Oracle Calendar Application from the Oracle User Portal.				
	A.	Login to the Oracle Collaboration Suite User Portal (as demonstrated in Subtest 1 of the Oracle User Login Test). The Oracle Collaboration Suite User Portal will be divided graphically into the following sections: `Links`, `News`, `Mail`, `Content Services`, `Web Conferencing`, `Tasks`, and `Calendar` Hit the 'Refresh' button on the browser if any of the sections are listed as 'Unavailable.'		
	B.	Click on the link labeled `Calendar` (which is below `Links`). A new browser window (the Calendar window) should appear with the text `Oracle Collaboration Suite Calendar` across the top of the page.		
	PASS criteria for Oracle Calendar Subtest 1: An additional window (the Content Services window) will appear, displaying the content described in Block B.		**P/F**	
2. Subtest 2: Set an appointment (meeting) on the user's calendar.				
	A.	In the Calendar Window (see Step 1.B. above), beneath the text `Oracle Collaboration Suite Calendar` will be a row of 10 icons. Below the icons will be the title `Daily View`. Click the 5th icon (from left) which is the image of a clock with a yellow plus sign (the text `Create a Meeting` appears when the cursor goes over the icon).		
	B.	The Calendar Window will load a new page, with the heading `New Meeting` at the top left of the page. The cursor will automatically be located in a text field to the left of the words `Title:`. In that text field, type `Test Meeting`. Click the `Create` button, and a new page will load (**Note:** By default, the Test Meeting will be 1 hour in duration, and be scheduled to begin at the start of the next hour).		
	C.	The Calendar Window will return to the `Daily View` page. Verify that the New Meeting appointment is now on the Oracle User's Calendar, for 1 hour duration.		
	D.	Click the link in the upper right corner labeled `Return to Portal`. The browser will return to the Oracle User Portal. Verify that the `Test Meeting` Appointment is now visible on the `Calendar` section of the Oracle User Portal.		
	PASS criteria for Oracle Calendar Subtest 2: The appointment made in Steps A through B will appear in the Oracle User's Calendar (as noted in Step C and D).		**P/F**	
Overall Pass/Fail? All Oracle Content Services (Web Browser) Subtests must be marked with a 'Pass.'			**P/F**	

E. TEST THE ORACLE WORKSPACES (WEB BROWSER) APPLICATION

The Oracle Workspaces (web browser) application will be accessed from the Oracle Collaboration Suite Portal using one of two web browsers: either the Internet Explorer web browser, or the Mozilla FireFox web browser. These procedures can be completed on either OCS Client 1 or OCS Client 2. All of the procedures listed in this section are to be completed under the *Administrator* account in Windows. Ensure that the pop-up blockers on Internet Explorer and Mozilla FireFox are disabled.

The procedures in this section verify that the Oracle Workspaces (web browser) application is functioning properly on the OCS 10g server. The three parts of this section of the Appendix are based on three individual tests: (1) the Web Browser test, (2) the Oracle User Login test, and (3) the Oracle Workspaces (web browser) test. The tests in this section should be completed in order, from the first to the last. If any one of the three tests is marked as a 'Fail,' it will be recorded that the Oracle Workspaces (web browser) application did not function as expected.

1. Web Browser Test

This test will verify that the web browser can (1) open correctly, and (2) access the web site http://ocsserver1.cisrlabmlstestbed3.com:80. Individual subtests (ex. Subtest 1, Subtest 2) will be marked 'P' for Pass, or 'F' for Fail. If one (or more) of the subtests of this test is marked as a Fail, the test will be recorded as a 'Fail' overall. Table 3 describes the details of this test procedure.

2. Oracle User Login Test

This test will verify that that an Oracle User can login to the Oracle Collaboration Suite User Portal via the web browser. The individual subtest (Subtest 1) will be marked 'P' for Pass, or 'F' for Fail. If the individual subtest of this test is marked as a Fail, the test will be recorded as a Fail overall. Table 4 describes the details of this test procedure.

3. Oracle Workspaces Application (Web Browser) Test

This test will verify that the Oracle Workspaces (web browser) application can (a) be accessed from the User Portal, and (b) create workspace for the user. Individual subtests (ex. Subtest 1, Subtest 2.) will be marked 'P' for Pass, or 'F' for Fail. If one (or more) of the subtests of this test is marked as a Fail, the test will be recorded as a ‘Fail’ overall. Table 9 describes the details of this test procedure.

Table 9. Oracle Workspaces Test

Oracle Workspaces (Web Browser) Tests: To verify that the Oracle Workspaces web browser application can (a) be accessed from the User Portal, and (b) create a workspace for the user. Individual subtests (ex. Subtest 1, Subtest 2) will be marked 'P' for Pass, or 'F' for Fail				
1. Subtest 1: Access the Oracle Workspaces Application from the Oracle User Portal.				
	A.	Login to the Oracle Collaboration Suite User Portal (as demonstrated in Subtest 1 of the Oracle User Login Test). The Oracle Collaboration Suite User Portal will be divided graphically into the following sections: `Links`, `News`, `Mail`, `Content Services`, `Web Conferencing`, `Tasks`, and `Calendar` Hit the 'Refresh' button on the browser if any of the sections are listed as 'Unavailable.'		
	B.	In the `Links` section, find and click on the icon labeled `Workspaces`. A new page will load in the browser window, with the text `Oracle Collaboration Suite Workspaces` across the top of the page.		
	PASS criteria for Oracle Workspaces Subtest 1: The Workspaces page will load in the window, as described in Block B.		**P/F**	
2. Subtest 2: Create an Oracle Workspace for the first Oracle User Account.				
	A.	In the browser window (see Step 1.B. above), find and click on the button labeled `New Workspace`.		
	B.	The browser window will display a new page with the heading `Select a workspace template and click Next.` at the top left of the page. The title `My Workspaces` will appear under this heading. By default, `Basic Workspace Template` will be selected. Find and click on the gray `Next` button on the right side of the page. A new page will load.		
	C.	The browser window will display a new page with the heading `New Workspace Using Template.` Left click the text field to the right of the heading `Workspace Name`, and type `Workspace Test 1.` Left click the text field to the right of the heading `Display Name`, and type `Workspace Test 1.` Click the grey `OK` button at the bottom right area of the page.		
	D.	The page titled `My Workspaces` will reload. Verify that the workspace just created is now listed under the group labeled `All Workspaces`.		
	PASS criteria for Oracle Workspaces Subtest 2: An Oracle Workspace can be created.		**P/F**	
Overall Pass/Fail? All Oracle Workspaces (Web Browser) Tests must be marked with a 'Pass.'			**P/F**	

F. TEST THE ORACLE DISCUSSIONS (WEB BROWSER) APPLICATION

The Oracle Discussions (web browser) application will be accessed from the Oracle Collaboration Suite Portal using one of two web browsers: either the Internet Explorer web browser, or the Mozilla FireFox web browser. These procedures can be completed on either OCS Client 1 or OCS Client 2. All of the procedures listed in this section are to be completed under the *Administrator* account in Windows. Ensure that the pop-up blockers on Internet Explorer and Mozilla FireFox are disabled. An Oracle Workspace should be created (as described in Section E) prior to conducting the procedures in Section F

The procedures in this section verify that the Oracle Discussions (web browser) application is functioning properly on the OCS 10g server. The three parts of this section of the Appendix are based on three individual tests: (1) the Web Browser test, (2) the Oracle User Login test, and (3) the Oracle Discussions (web browser) test. The tests in this section should be completed in order, from the first to the last. If any one of the three tests is marked as a 'Fail,' it will be recorded that the Oracle Discussions (web browser) application did not function as expected.

1. Web Browser Test

This test will verify that the web browser can (1) open correctly, and (2) access the web site http://ocsserver1.cisrlabmlstestbed3.com:80. Individual subtests (ex. Subtest 1, Subtest 2) will be marked 'P' for Pass, or 'F' for Fail. If one (or more) of the subtests of this test is marked as a Fail, the test will be recorded as a 'Fail' overall. Table 3 describes the details of this test procedure.

2. Oracle User Login Test

This test will verify that that an Oracle User can login to the Oracle Collaboration Suite User Portal via the web browser. The individual subtest (Subtest 1) will be marked 'P' for Pass, or 'F' for Fail. If the individual subtest of this test is marked as a Fail, the test will be recorded as a Fail overall. Table 4 describes the details of this test procedure.

3. Oracle Discussions Application (Web Browser) Test

This test will verify that the Oracle Discussions (web browser) application can (a) be accessed from the User Portal, and (b) create workspace for the user. Individual subtests (ex. Subtest 1, Subtest 2.) will be marked 'P' for Pass, or 'F' for Fail. If one (or more) of the subtests of this test is marked as a Fail, the test will be recorded as a ‘Fail’ overall. Table 10 describes the details of this test procedure.

Table 10. Oracle Discussions Test

Oracle Discussions (Web Browser) Tests: To verify that the Oracle Workspaces web browser application can (a) be accessed from the User Portal, and (b) create a discussion thread. Individual subtests (ex. Subtest 1, Subtest 2) will be marked 'P' for Pass, or 'F' for Fail. **Note:** Ensure that the Oracle Workspace (Web Browser) Test has been completed prior to conducting this test.			
1. Subtest 1: Access the Oracle Workspaces Application from the Oracle User Portal.			
	A.	Login to the Oracle Collaboration Suite User Portal (as demonstrated in Subtest 1 of the Oracle User Login Test). The Oracle Collaboration Suite User Portal will be divided graphically into the following sections: `Links`, `News`, `Mail`, `Content Services`, `Web Conferencing`, `Tasks`, and `Calendar` Hit the 'Refresh' button on the browser if any of the sections are listed as 'Unavailable.'	
	B.	In the `Links` section, find and click on the icon labeled `Discussions`. A new page will load in the browser window, with the text `Oracle Collaboration Suite Discussions` across the top of the page.	
	PASS criteria for Oracle Discussions Subtest 1: The Discussions page will load in the window, as described in Block B.	**P/F**	
2. Subtest 2: Create an Oracle Workspace for the first Oracle User Account.			
	A.	In the browser window, find and click the link labeled `Workspace_forums`.	
	B.	The new page will include the title `Category: Workspace_forums`. Under the title, locate and click on the link labeled `WORKSPACE TEST 1`. A new page will load.	
	C.	Under the title `Category: WORKSPACE TEST 1`. click on the `New Forum` button.	
	D.	The heading `New Forum` will appear at the top of the page. Left click the text field to the right of the heading `New Forum Name`, and type `Discussions Test 1.` Left click the text field to the right of the heading `Forum Display Name, and type Discussions Test 1.` Click the grey `Done` button. A new page will load.	
	D.	The words `Confirmation... Forum "Discussions Test 1" has successfully been created.` will appear. Under the heading `Category: WORKSPACE TEST 1,` Left click the `Discussion Test 1` link.	
	E.	Under the heading `Forum: Discussion Test 1`, click the button labeled `New Topic`. A page with the heading `New Topic` will appear. Left click the text field to the right of the heading `Subject` and type `Test 1 Thread`. Left click the large text field below and type `The Quick Brown Fox Jumps Over the Lazy Dog.` Click the grey `Post` button.	
	D.	Verify that the `Forum: Discussion Test 1` page displays `Confirmation Forum "Test 1 Thread" has successfully been created`.	
	PASS criteria for Oracle Discussions Subtest 2: An Oracle Discussions thread can be created in workspace `Workspace Test 1`.	**P/F**	
Overall Pass/Fail? All Oracle Discussion (Web Browser) Tests must be marked with a 'Pass.'		**P/F**	

G. TEST THE ORACLE RTC INSTANT MESSENGER (RICH MEDIA CLIENT) APPLICATION

The procedures in this section verify that the Oracle RTC Instant Messenger rich media client application is functioning properly on the OCS 10g server. The RTC Instant Messenger is a downloadable plug in client that provides a chat interface for Oracle Users. The first part of this section details how to download and install the Oracle RTC Instant Messenger from the Oracle Collaboration Suite portal. The second part is the Oracle RTC Instant Messenger (rich media client) test. The parts in this section should be completed in order, from the first to the last. If either (a) the Oracle RTC Messenger cannot be downloaded and installed, or (b) the Oracle RTC Instant Messenger test is marked as a 'Fail,' it will be recorded that the Oracle RTC Instant Messenger application did not function as expected.

The Oracle (Real-Time Collaboration) RTC Instant Messenger rich media client application will be downloaded from the Oracle Collaboration Suite Portal, and installed on both OCS Client 1 and OCS Client 2 (these procedures can be completed on either OCS Client 1 or OCS Client 2). All of the procedures listed in this section are to be completed under the *Administrator* account in Windows. Ensure that the pop-up blockers on Internet Explorer and Mozilla FireFox are disabled.

1. Install the Oracle RTC Messenger

This test will consist of one subtest, Subtest 1: Download and Install the Oracle RTC Messenger. If the individual subtest of this test is marked as a Fail, the test will be recorded as a Fail overall.

a. Test 1: Download and Install the Oracle RTC Messenger

Step 1. Using either Mozilla FireFox or Internet Explorer, access the web site http://ocsserver1.cisrlabmlstestbed3.com:80. The browser window should change to the heading 'End-User Resources.' The greeting 'Welcome to Oracle Collaboration Suite' will appear under the title 'ORACLE Collaboration Suite.'

Step 2. Click the link titled ‘Oracle Desktop Access’ which lies underneath the heading ‘Downloads’ on the right side of the page.

Step 3. ‘Download Oracle Desktop Clients and Tools’ will appear under the title ‘ORACLE Collaboration Suite.’ Under the heading ‘Oracle Messenger,’ click the link titled ‘Windows.’

Step 4. A warning message stating ‘Opening setup.exe… Would you like to save this file?’ will appear. Click ‘Save file.’ **Note:** If you are using the Internet Explorer (IE) web browser, the message may appear differently. If using IE, click the button labeled ‘OK.’

Step 5. A second warning message will appear, stating ‘Open Executable File?’ Click the ‘OK’ button.

Step 6. A small window titled ‘Oracle Messenger Setup’ will appear, asking ‘Install Oracle Messenger?’ Click ‘Yes.’

Step 7. Once the RTC Oracle Messenger completes installation, a new window will appear (the Oracle Messenger window) will automatically popup, and the RTC Messenger will attempt to sign in to the OCS 10g server. Annotate that the Oracle Messenger was correctly installed by marking a ‘P’ for Pass in the blank below Step 8.

Step 8. Leave the Oracle Messenger Window open, and proceed to the next part of this section.

P/F _____

2. Oracle RTC Instant Messenger (rich media client) Test

This test will verify that the Oracle RTC Instant Messenger application can and (a) connect to the server, and (b) send a chat message. Individual subtests (ex. Subtest 1, Subtest 2.) will be marked 'P' for Pass, or 'F' for Fail. If one (or more) of the subtests of this test is marked as a Fail, the test will be recorded as a ‘Fail’ overall.

a. Test 1: Connect the RTC Instant Messenger to the OCS Server

Step 1. Startup the Oracle RTC Instant Messenger by double clicking on the Oracle Messenger icon on the client's desktop. **Note:** If the Oracle RTC Instant Messenger is already running, skip this step and proceed to Step 2.

Step 2. Verify the connection to the server: if the RTC Instant Messenger connects to the server, and a pop-up window stating 'sign in failure message' does not appear, annotate that the Oracle Messenger connected to the server by marking a 'P' for Pass in the blank below. **Note:** If you receive a 'sign in failure' message, select 'Tools' on the menubar of the Oracle Messenger window, and select Options from the drop-down box that appears. Another window will appear, with the title 'Options' at the top. On the white box on the left side of this window, click on 'Connections.' Under the 'Connections' subtitle that appears on the right side, verify that (a) the radio button for 'Automatic Configuration for RTC Connection is checked, (b) HTTP is selected, (c) the text field to the right of 'Web Host' reads http://ocsserver1.cisrlabmlstestbed3.com, and (d) the text field to the right of 'Web Port' reads '80.' Then click 'OK' at the bottom right corner of the 'Options' window. If this does not establish a connection to the OCS 10g server, annotate that the Oracle Messenger could not connect to the server by marking 'F' for Fail in the blank below.

P/F _____

b. Test 2: Send a Chat Message using the RTC Instant Messenger

Step 1. In the window titled 'Oracle Messenger,' click on the link labeled 'Chat.'

Step 2. A new window titled ‘Oracle Messenger: Chat Conferencing’ will appear. Click on ‘Actions’ on the menubar, and select ‘Start New Chat Conference’ on the drop-down box that appears.

Step 3. A new window titled ‘Oracle RTC Messenger: Start a Conference’ will appear. Click on the text field directly below the ‘Conference Title:’ heading, and type ‘RTC Instant Messenger Test 1.’ Click the ‘Send’ button.

Step 4. The ‘Oracle Messenger: Chat Conferencing’ window will become a chat interface window. Verify that the words ‘Title: RTC Instant Messenger Test 1’ appears just below the chat response box on the left side of the window. Click on the large text field below the ‘Title’ heading, and type ‘The Quick Brown Fox Jumps Over the Lazy Dog.’ Verify that the above response has been posted.

Step 5. Annotate that the Oracle Messenger could send a chat message by marking a ‘P’ for Pass in the blank below.

P/F _____

H. TEST THE ORACLE CONTENT SERVICES ‘ORACLE DRIVE’ APPLICATION

The procedures in this section verify that the Oracle Content Services ‘Oracle Drive’ rich media client application is functioning properly on the OCS 10g server. The Content Services ‘Oracle Drive’ is a downloadable plug in client that provides a WebDAV-based application for Oracle Users. The first part of this section details how to download and install the Oracle Content Services ‘Oracle Drive’ from the Oracle Collaboration Suite portal. The second part is the Oracle Content Services ‘Oracle Drive’ (rich media client) test. These parts should be completed in order. If either (a) the Oracle Content Services ‘Oracle Drive’ application cannot be downloaded and installed, or (b) the Oracle Content Services ‘Oracle Drive’ test is marked as a ‘Fail,’ it will be recorded that the Oracle Content Services ‘Oracle Drive’ application did not function as expected.

The Oracle (Real-Time Collaboration) Content Services 'Oracle Drive' rich media client application will be downloaded from the Oracle Collaboration Suite Portal, and installed on both OCS Client 1 and OCS Client 2 (these procedures can be completed on either OCS Client 1 or OCS Client 2). All of the procedures listed in this section are to be completed under the *Administrator* account in Windows. Ensure that the pop-up blockers on Internet Explorer and Mozilla FireFox are disabled.

1. Install the Oracle Content Services 'Oracle Drive'

This test will consist of one subtest, Subtest 1: Download and Install the Oracle Content Services 'Oracle Drive.' If the individual subtest is marked as a Fail, the test will be recorded as a Fail overall.

a. Test 1: Download and Install the Oracle Content Services 'Oracle Drive'

Step 1. Using either Mozilla FireFox or Internet Explorer, access the web site http://ocsserver1.cisrlabmlstestbed3.com:80. The browser window should change to the heading 'End-User Resources.' The greeting 'Welcome to Oracle Collaboration Suite' will appear under the title 'ORACLE Collaboration Suite.'

Step 2. Click the link titled 'Oracle Desktop Access' which lies underneath the heading 'Downloads' on the right side of the page.

Step 3. The browser window that opens should be titled 'Download Oracle Desktop Access Components.' The greeting 'Download Oracle Desktop Clients and Tools' will appear under the title 'ORACLE Collaboration Suite.' Under the heading 'Oracle Drive,' click the link titled 'Windows.'

Step 4. A warning message stating 'Opening ODriveSetup.exe… Would you like to save this file?' will appear. Click 'Save file.' **Note:** If

you are using the Internet Explorer (IE) web browser, the message may appear differently. If using IE, click the button labeled ‘OK.’

Step 5. A second warning message will appear, stating ‘Open Executable File?’ Click the ‘OK’ button.

Step 6. A small window titled ‘Choose Language Setting’ will appear. Select ‘English: United States.’ Click ‘OK.’

Step 7. A larger window titled ‘Oracle Drive 10.2.1.2-InstallShield Wizard’ will appear. Click the ‘Next’ button.

Step 8. Click the ‘Next’ button again.

Step 9. The text ‘Ready to Install the Program’ will appear at the top left of the window. Click the ‘Install’ button.

Step 10. Once installation is complete, check the box marked ‘Place Oracle Drive Shortcut onto Desktop,’ and click the ‘Next’ button.

Step 11. Select the radio button labeled ‘Yes, I want to restart my computer now,’ and click the ‘Finish’ button (the computer will restart automatically).

Step 12. Following restart, login under the Windows *Administrator* account. Verify that the ‘Oracle Drive’ shortcut is now displayed on the client’s desktop. Annotate that the Oracle Content Services ‘Oracle Drive’ was correctly installed by marking a ‘P’ for Pass in the blank below.

P/F _____

2. Oracle Content Services ‘Oracle Drive’ (rich media client) Test

This test will verify that the Oracle Content Services ‘Oracle Drive’ application can and connect to the server and access the /cisrlabmlstesbed3 directory. This test will consist of one subtest, Subtest 1: Connect the Oracle Drive to the OCS 10g server and Access the /cisrlabmlstestbed3 Directory. If the individual subtest is marked as a Fail, the test will be recorded as a Fail overall.

a. Test 1: Connect the Oracle Drive to the OCS 10g Server and Access the /cisrlabmlstestbed3 Directory

Step 1. Startup the Oracle Content Services 'Oracle Drive' by double clicking on the 'Oracle Drive' icon on the client's desktop. **Note:** If the Oracle Content Services 'Oracle Drive' is already running, skip this step and proceed to Step 2.

Step 3. A window titled 'Oracle Drive' will appear. Click in the text field next to the heading 'Username:', and enter the user name of the first Oracle User Account (ex. 'jpjones').

Step 4. Click the radio button labeled 'Service,' and on the drop-down box that appears, select 'New.' In the window that appears (Titled 'Service Properties,' type the user name of the first Oracle User Account into the text field below the heading 'Username.' In the text field below the heading 'Sever,' type 'ocsserver1.cisrlabmlstestbed3.com.' Click the 'OK' button.

Step 5. Click the radio button labeled 'Advanced.' Verify that the 'Port' is set to '80,' and the 'server directory' is set to 'content/dav.' **Note:** If directly connected to OCS Server, then check the box 'Bypass proxy server for this connection.'

Step 6. On the menubar, click the tab labeled 'Options.' Under 'Options,' click the button labeled 'Change Proxy Settings.'

Step 7. A new window will appear with the title 'Proxy Settings.' **Note:** If the OCS 10g server is directly connected to the OCS Clients, make sure that the gray fields on the top right side *both* read "Direct connection, no proxy." If the OCS Clients are using the XTS-400 as a proxy, (a) check the box labeled 'HTTP,' (b) enter '192.168.0.130' in the field directly under 'Proxy Server,' (c) uncheck the 'Secure' box until 'Auto-detect on connection' appears in the text field across from 'Secure,' (d) under the 'HTTP Proxy Authentication,' enter 'mdemo1' or 'mdemo2'

in the text field corresponding to ‘User name’ (see Appendix A, Section N for an explanation regarding which username to use) and type ‘tcxuser’ in the ‘Password’ text field. Click the ‘OK’ button.

Step 8. On the ‘Oracle Drive’ window, click the ‘Connect’ tab on the menubar. Find the radio button labeled ‘Connect’ and click it.

Step 9. An Internet Explorer browser window should open, containing a single file folder named ‘/cisrlabmlstesbed3.’ Annotate that the Oracle Content Services could access the Oracle directory ‘/cisrlabmlstestbed3.com’ by marking a ‘P’ for Pass in the blank below

P/F _____

I. TEST THE ORACLE SMTP MAIL SERVER

The procedures in this section verify that the Oracle SMTP Mail Server can be reconfigured to send an Oracle Web Mail email from an Oracle User on the OCS 10g server to an email address on the Linux Mail server preconfigured by the CISR staff (described in Appendix A). The Content Services ‘Oracle Drive’ is a downloadable plug in client that provides a WebDAV-based application for Oracle Users. The first part of this section details how to configure the `SMTP_inbound and SMTP_outbound` settings on the Oracle Collaboration Suite Application Control Console to exchange email with the Linux mail server. The second part is an email test from the OCS 10g server to the Linux mail server, using the Oracle Web Mail application. If either (a) the SMTP settings cannot be configured as described in the first part, or (b) an email cannot be sent from the OCS 10g server to the Linux mail server, the Oracle SMTP Mail Server test is marked as a ‘Fail,’ and it will be recorded that the Oracle SMTP Mail Server could not be reconfigured to exchange email with another external mail server at this time.

All of the procedures listed in this section are to be completed under the *Administrator* account in Windows. Ensure that the pop-up blockers on Internet Explorer and Mozilla FireFox are disabled.

1. Modify the SMTP Settings

This test will consist of one subtest, Subtest 1: Change SMTP settings on the Oracle Application Control Console. If the individual subtest is marked as a Fail, the test will be recorded as a Fail overall.

a. Test 1: Change SMTP Settings on the Oracle Application Control Console

Step 1. Using either Mozilla FireFox or Internet Explorer, access the web site http://ocsserver1.cisrlabmlstestbed3.com:18100. A separate window appears, stating 'The server ocsserver.cisrlabmlstestbed3.com at enterprise-manager is asking for a password.' Enter `ias_admin` for the user name, and `password123` (or whatever has been selected as the `ias_admin` password) for the password.

Step 2. The browser window should open to a page titled 'Oracle Enterprise Manager-Farm:orcl.cisrlabmlstestbed3.com.' The heading 'Enterprise Manager 10g, Application Server Control' will appear at the top left of the screen. Below this heading will be the text 'Farm: orcl.cisrlabmlstestbed3.com.' Under 'Standalone Instances,' click the link labeled 'ocsapps.ocsserver1.cisrlabmlstestbed3.com.' **Note:** A separate window might appear again, stating 'The server ocsserver.cisrlabmlstestbed3.com at enterprise-manager is asking for a password.' Enter `ias_admin` for the user name, and `password123` (or whatever has been selected as the `ias_admin` password) for the password.

Step 3. The browser window should reload the page titled 'Oracle Enterprise Manager- Application Server: ocsapps.ocsserver1.cisrlabmlstestbed3.com.' The heading 'Enterprise Manager 10g, Application Server Control for Collaboration Suite' will

appear at the top left of the screen. Below this heading will be the text ‘Application Server Control for Collaboration Suite: ocsapps.ocsserver1.cisrlabmlstestbed3.com.’ Under the blue text ‘System Components,’ locate and click on the link labeled ‘Mail Application.’

Step 4. The browser window should open to a page titled ‘Oracle Enterprise Manager- Mail Application.’ The heading 'Enterprise Manager 10g, Application Server Control for Collaboration Suite’ will appear at the top left of the screen. Under the blue text ‘Service Targets,’ locate and click on the link labeled ‘SMTP Inbound Server.’

Step 5. The browser window should open to a page titled ‘Oracle Enterprise Manager- SMTP Inbound Server.’ The heading 'Enterprise Manager 10g, Application Server Control for Collaboration Suite’ will appear at the top left of the screen. Under the blue text ‘Process Instances,’ locate and click on the link labeled ‘smtp_in:…..’ (the ….. represents a random number that the OCS 10g server has assigned to the `SMTP_inbound` application).

Step 6. The browser window should open to a page titled ‘Oracle Enterprise Manager- smtp_in:…..’ The heading 'Enterprise Manager 10g, Application Server Control for Collaboration Suite’ will appear at the top left of the screen, with the words ‘Mail Collaboration Suite Database’ under the heading Scroll down on the page and left click the text field to the left of the words ‘Trusted Relay Domains.’ Type ‘cisrlabmlstestbed3.com’ in the text field, scroll down to the bottom of the page, and click on the button labeled ‘Apply.’

Step 7. The page titled ‘Oracle Enterprise Manager- Status’ will load, with the following blue text: ‘Confirmation: The process settings have been modified successfully.’ On the group of links above this text, find and click the link labeled ‘Mail Application.’

Step 8. The browser window should refresh to the page titled ‘Oracle Enterprise Manager- Mail Application.’ Under the blue text ‘Service Targets,’ locate and click on the link labeled ‘SMTP Outbound Server.’

Step 9. The browser window should refresh to a page titled ‘Oracle Enterprise Manager- SMTP Outbound Server.’ The heading 'Enterprise Manager 10g, Application Server Control for Collaboration Suite’ will appear at the top left of the screen. Under the blue text ‘Process Instances,’ locate and click on the link labeled ‘smtp_out:…..’ (the ….. represents a random number that the OCS 10g server has assigned to the `SMTP_outbound` application).

Step 10. The browser window should refresh to a page titled ‘Oracle Enterprise Manager- smtp_out:…..’ The heading 'Enterprise Manager 10g, Application Server Control for Collaboration Suite’ will appear at the top left of the screen, with the words ‘Mail Collaboration Suite Database’ under the heading Scroll down on the page and left click the text field to the left of the words ‘SMTP Relay.’ Type ‘cisrlabmlstestbed3.com’ in the text field, scroll down to the bottom of the page, and click on the button labeled ‘Apply.’

Step 11. The page titled ‘Oracle Enterprise Manager- Status’ will load, with the following blue text: ‘Confirmation: The process settings have been modified successfully.’ On the group of links above this text, find and click the link labeled ‘Application Server: ocsapps.ocsserver1.cisrlabmlstestbed3.com.’

Step 12. The browser window should reload the page titled ‘Oracle Enterprise Manager- Application Server: ocsapps.ocsserver1.cisrlabmlstestbed3.com.’ Under the blue text ‘System Components,’ locate the link labeled ‘Mail Application,’ and check the

box located immediately to the left of the link. Under the 'System Components' tab, click the 'Reload' button.

Step 13. The page that loads will contain the text,' You have chosen to reload the Mail Application. Do you wish to proceed?' Click 'Yes.'

Step 14. The page that loads will state, 'Application restart in progress.. please wait.' Once the application has been reloaded, mark a 'P' for Pass in the blank below

P/F _____

2. Send Email from Oracle Web Mail to Account on Linux Mail Server

This test will verify that the OCS 10g server can send an email via SMTP to an email account on another Linux mail server. This test consists of two subtest, Subtest 1: Access the Oracle Web Mail Application from the Oracle User Portal, and Subtest 2: Send an email from the OCS 10g server to the Linux mail server. Individual subtests (ex. Subtest 1, Subtest 2) will be marked 'P' for Pass, or 'F' for Fail. If one (or more) of the subtests of this test is marked as a Fail, the test will be recorded as a 'Fail' overall. Table 11 describes the details of this test procedure.

Table 11. Oracle SMTP External Mail Server Tests

Oracle SMTP External Mail Server Tests: To verify that the Oracle Web Mail web browser application can (a) open Web Mail from the User Portal, and (b) send an email from the OCS 10g server to an email account on a Linux mail server located on the same domain (`cisrlabmlstestbed3.com`). Individual tests (ex. Test 1., Test 2.) will be marked 'P' for Pass and 'F' for Fail.		
1. Test 1: Access the Oracle Web Mail Application from the Oracle User Portal.		
	A.	Login to the Oracle Collaboration Suite User Portal (as demonstrated in Test 1 of the Oracle User Login Test). The Oracle Collaboration Suite User Portal will be divided graphically into the following sections: `Links`, `News`, `Mail`, `Content Services`, `Web Conferencing`, `Tasks`, and `Calendar`. Hit the 'Refresh' button on the browser if any of the sections are listed as 'Unavailable.'
	B.	Click on the link labeled `Mail` (which is below `News` and above `Content Services`). A new browser window (the Web Mail window) should appear with the an image of an envelope and the word `MAIL` immediately to the left of the image.

	PASS criteria for Oracle SMTP External Mail Server Test 1: An additional window (the Web Mail window) will appear, displaying the content described in Block B.		**P/F**	
2. Test 2: Send an email to an email account on the Linux mail server using Oracle Web Mail.				
	A.	In the Web Mail Window (see Step 1.B. above), the following headings will appear above the `MAIL` image, from left to right: `New, View, Go, Actions, Print, Delete, and Find People.` Click the heading labeled `New`, and a new browser window will appear (the New		
	B.	In the New Message Window, click the heading labeled `Format`, and on the drop down box that appears, make sure that `HTML` is selected.		
	C.	Click the text field to the right of the `To:` button, and type in an email address of an account on the Linux mail server (**Note:** this email address will be provided by the CISR staff).		
	D.	Click the text field to the right of the `Subject` heading, and type `Email Test.`		
	E.	Click the large text field immediately below `Subject` and type `Test SMTP.` Click the `Send` button (the New Message window will close).		
	F.	On the Web Mail Window, click the `Sent Items` heading. The large title to the right should now read `Sent Items.` The field below this title should contain the email that was sent in Step D (titled `Email Test`).		
	PASS criteria for Oracle SMTP External Mail Server Test 2: The Oracle Webmail constructed in Steps A through E will appear in the `Sent Items` folder (as noted in Step F).		**P/F**	
Overall Pass/Fail? All Oracle Web Mail Tests must be marked with a 'Pass.'			**P/F**	

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERNCES

[1] USN Program Executive Office for Command, Control, Communications, Computers and Intelligence (PEO C4I), "Net-Centric Implementation Framework, Part 1: Overview, Version 2.1.0," 12 October 2007.

[2] Defenselink.com, "Definition: Business Enterprise Architecture," Available: http://www.defenselink.mil/DBT/faq_bea.html [Accessed 13 February 2008].

[3] Wikipedia.com. "Service-oriented Architecture," Available: http://en.wikipedia.org/wiki/Service-oriented_architecture [Accessed 10 February 2008].

[4] P. Strassman, "Service-Oriented Architecture (SOA) for DoD," 9 January 2008, Lecture slides, George Mason University.

[5] B. Bradley, "Business Transformation at the Department of Defense," *Cio.com,* July 6, 2007, Available: http://www.cio.com/article/print/122605 [Accessed 12 February 2008].

[6] R. Maule, "TACFIRE Trident Warrior 07 MDA (HA/DR) in MHQ with MOC Operations," 24 July 2007.

[7] C. E. Irvine, T. E. Levin, T. D. Nguyen, D. Shifflett, J. Khosalim, P. C. Clark, A. Wong, F. Afinidad, D. Bibighaus, J. Sears, "Overview of a High Assurance Architecture for Distributed Multilevel Security," in *2004 IEEE Systems, Man and Cybernetics Information Assurance Workshop*, West Point, NY, June 2004.

[8] T. D. Nguyen, T. E. Levin, C. E. Irvine, "MYSEA Testbed," *Proceedings of the 2004 IEEE Systems, Man and Cybernetics Information Assurance Workshop,* West Point, NY, June 2005, pg. 438-439.

[9] D. E. Bell, and L. J. LaPadula, "Secure Computer Systems: Mathematical Foundations." *MITRE Corporation Technical Report 2547*, vol. 1, 1973. Available: http://www.albany.edu/acc/courses/ia/classics/belllapadula1.pdf [Accessed 12 February 2008].

[10] A. Turban, Liang, & Sharda. *Decision Support and Business Intelligence Systems*, Upper Saddle River: Prentice Hall, 7th Edition, 2005.

[11] D. Krafzig, K. Banke, D. Slama, *Enterprise SOA: Service-Oriented Architecture Best Practices (The Coad Series),* Upper Saddle River: Prentice Hall, 1st Edition, June 2007.

[12] H. J. Watson, B. H. Wixon, "The Current State of Business Intelligence," *IT Systems Perspectives Magazine*, September 2007.

[13] InfoWorld.com, *SOA Report 2007: Service-Oriented Architecture Graduates to the Enterprise*, in *InfoWorld online*, July 2007.

[14] T. Ely, *Service-Oriented Architecture*, Upper Saddle River: Prentice Hall, 1st Edition, November 2004.

[15] R. C. Barnard, "Pentagon Takes Major Step Toward Battlespace System Harmonization," in *Sea Power online*, July 2004, Available: http://findarticles.com/p/articles/mi_qa3738/is_200407/ai_n9458648/pg_2 [Accessed 5 March 2008].

[16] Wikipedia.com. "Command Post of the Future", Available: http://en.wikipedia.org/wiki/Command_Post_of_the_Future [Accessed 4 February 2008].

[17] Defenselink.com, "Annual Report to the Congressional Defense Committees: Status of the Department of Defense's Business Transformation Efforts," 15 March, 2007, Available: http://www.defenselink.mil/dbt/products/2007_BEA_ETP/etp/Data/March_07_ETP_CR.pdf [Accessed 12 February 2008].

[18] United States Government Accountability Office online, "GAO Report to Congressional Addressees: DoD Needs to Ensure Navy Marine Corps Intranet is Meeting Goals and Satisfying Customers," December, 2006, Available http://www.gao.gov/new.items/d0751.pdf [Accessed 10 January 2008].

[19] H. He, "What is Service Oriented Architecture." Published on xml.com. 30 September 2003. Available http://webservices.xml.com/lpt/a/1292 [Accessed 18 January 2008].

[20] R. Maule, and S. Gallup, "TACFIRE: Enterprise knowledge in service-oriented architecture," *Proceedings of the IEEE/AFCEA Military Communications Conference*, Washington, DC, 23-25 October 2006.

[21] R. Maule, "Enterprise knowledge security architecture for military experimentation," *Proceedings of the IEEE SMC 2005 International Conference on Systems, Man and Cybernetics*, Waikoloa, HI, 10 October 2005.

[22] R. Maule, and S. Gallup, "FORCEnet Innovation & Research Enterprise (F.I.R.E.) architecture for computer supported cooperative work," *Proceedings of the 2007 International Conference on Software Engineering Theory and Practice (SETP-07)*, Orlando, FL, July 2007.

[23] R. Maule, "Quality of service assessment in SOA synchronous networked communications," *Proceedings of the 2007 International Conference on Computing, Communications and Control Technologies (CCCT 2007)*, Orlando, FL, 12-15 July 2007.

[24] R. Maule, and S. Gallup, "Knowledge management system with enterprise ontology for military experimentation – case study: F.I.R.E. and TACFIRE," *Proceedings of the 2007 International Conference on Enterprise Information Systems & Web Technologies (EISWT-07)*, Orlando, FL, July 2007.

[25] R. Maule, and S. Gallup, "Quality of service in next-generation knowledge management," *Proceedings of the International i-Society Conference*, Orlando, FL., May 2006.

[26] Oracle Technical Staff, *Collaboration Suite Concepts Guide, 10g Release 1 (10.1.2)*, Oracle document #B25491-03, July 2006.

[27] Oracle Technical Staff, *Collaboration Suite Installation Guide 10g Release 1 (10.1.2) for Windows*, Oracle document #B25463-03, September 2006.

[28] Oracle Technical Staff, *Oracle Collaboration Suite Deployment Guide 10g Release 1 (10.1.2)*, Oracle document #B25492-04, August 2006.

[29] Oracle.com, "Oracle Application Server 10g J2EE and Web Services: An Oracle White Paper," Oracle White Paper, August 2005 Available: http://www.oracle.com/technology/tech/java/oc4j/1012/collateral/OC4J-TWP-101202.pdf [Accessed 12 February 2008].

[30] Oracle Technical Staff, *Oracle Real Time Collaboration Administrator's Guide 10g Release 1 (10.1.2)*, Oracle document #B25460-03, May 2006.

[31] J. Bradney, "Use of WebDav to Support A Virtual File System In A Coalition Environment," M.S. Thesis, Naval Postgraduate School, Monterey, CA. June 2006.

[32] Oracle Technical Staff, *Oracle Collaboration Suite Administrator's Guide 10g Release 1 (10.1.2) for Windows or UNIX*, Oracle document #B25490-05, June 2006.

[33] Oracle Technical Staff, *Oracle Collaboration Suite Migration and Coexistence Guide 10g Release 1 (10.1.2) for Windows or UNIX*, Oracle document #B25493-03, April 2006.

[34] R. Maule, "RE: OCS Single Server Application Components," (private conversation), February, 14, 2008.

[35] Oracle Technical Staff, *Oracle Collaboration Suite 10g Datasheet*, June 2006.

[36] P. Henty, "Oracle Fusion Middleware R11: Statement of Direction," Oracle Web Center Organization, July, 2007.

[37] G. Pavlik, "Next-Generation SOA Infrastructure: An Oracle White Paper," Oracle White Paper, May, 2007.

[38] P. A. Buxbaum, "Web of Tomorrow," in *MIT 11.5*, Available: http://www.mit-kmi.com [Accessed 30 September 2007].

[39] Oracle Technical Staff, *Oracle Mail Administrator's Guide 10g Release 1 (10.1.2)*, Oracle document #B25499, April 2006.

[40] R. Maule, "Re: RTC legacy tools" (private communication), 28 January 2008.

[41] Jonathon B. Postel, "RFC 821: Simple Mail Transfer Protocol," IETF Network Working Group, August 1982, Available: http://www.ietf.org/rfc/rfc0821.txt [Accessed 12 February 2008].

[42] P. Saint-Andre, "RFC 3920: Extensible Messaging and Presence Protocol (XMPP)," IETF Network Working Group, September 2004, Available: http://tools.ietf.org/rfc/rfc3920.txt [Accessed 12 February 2008].

[43] Jabber.org, "Overview of Jabber," Jabber Software Foundation, 2005, Available: http://www.jabber.org/about/overview.shtml [Accessed 13 February 2008].

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, VA

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, CA

3. Susan Alexander
   OASD/NII DOD/CIO
   Washington, DC

4. Hugo A. Badillo
   NSA
   Fort Meade, MD

5. George Bieber
   OSD
   Washington, DC

6. Dr. Dan Boger
   Naval Postgraduate School
   Monterey, CA

7. Ed Bryant
   Unified Cross Domain Management Office
   Maryland

8. John Campbell
   National Security Agency
   Fort Meade, MD

9. Dr. Grace Crowder
   NSA
   Fort Meade, MD

10. Louise Davidson
    National Geospatial Agency
    Bethesda, MD

11. Steve Davis
NRO
Chantilly, VA

12. Vincent J. DiMaria
National Security Agency
Fort Meade, MD

13. Boyd Fletcher
SPAWAR
San Diego, CA

14. Dr. Tim Fossum
National Science Foundation

15. LTJG Craig Gilkey
Naval Postgraduate School
Monterey, CA

16. Jennifer Guild
SPAWAR
Charleston, SC

17. Mike Harrison
SPAWAR
San Diego, CA

18. Scott D. Heller
SPAWAR
Charleston, SC

19. Steve Iatrou
Naval Postgraduate School
Monterey, CA

20. Dr. Cynthia E. Irvine
Naval Postgraduate School
Monterey, CA

21. Keith Jarren
NSA
Fort Meade, MD

22. Steve LaFountain
NSA
Fort Meade, MD

23. Dr. Greg Larson
IDA
Alexandria, VA

24. Dr. Karl Levitt
NSF
Arlington, VA

25. Paul A. Livingston
Unified Cross Domain Management Office
Maryland

26. Dr. John Monastra
Aerospace Corporation
Chantilly, VA

27. John Mildner
SPAWAR
Charleston, SC

28. CDR James Mills
Joint Data Strategy Division (J87)
U.S. Joint Forces Command
Norfolk, VA

29. Randy W. Maule
Naval Postgraduate School
Monterey, CA

30. John P. Mcgeehan
Unified Cross Domain Management Office
Maryland

31. Thuy D. Nguyen
Naval Postgraduate School
Monterey, CA

32. Lt. Col. Karl Pfeiffer
Naval Postgraduate School
Monterey, CA

33. Charles Prince
Naval Postgraduate School
Monterey, CA

34. Mark T. Powell
Federal Aviation Administration
Washington, DC

35. Jim Roberts
Central Intelligence Agency
Reston, VA

36. Ed Schneider
IDA
Alexandria, VA

37. Mark Schneider
NSA
Fort Meade, MD

38. Keith Schwalm
Good Harbor Consulting, LLC
Washington, DC

39. Ken Shotting
NSA
Fort Meade, MD

40. CDR Wayne Slocum
SPAWAR
San Diego, CA

41. Matt Warnock
Booze-Allen-Hamilton

42. Dr. Ralph Wachter
ONR
Arlington, VA